Titre: Title:	Extension of the Overset Grid Preprocessor for Surface Conforming Meshes
Auteur: Author:	Julien Guay
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Guay, J. (2017). Extension of the Overset Grid Preprocessor for Surface Conforming Meshes [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2525/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2525/
Directeurs de recherche: Advisors:	Éric Laurendeau
Programme: Program:	Génie aérospatial

UNIVERSITÉ DE MONTRÉAL

EXTENSION OF THE OVERSET GRID PREPROCESSOR FOR SURFACE CONFORMING MESHES

JULIEN GUAY
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE AÉROSPATIAL)
AVRIL 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

EXTENSION OF THE OVERSET GRID PREPROCESSOR FOR SURFACE CONFORMING MESHES

présenté par : GUAY Julien
en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. CAMARERO Ricardo, Ph. D., président
M. LAURENDEAU Éric, Ph. D., membre et directeur de recherche
M. CASTONGUAY Patrice, Ph. D., membre

ACKNOWLEDGMENTS

My acknowledgments go first and foremost to my research director, Professor Eric Laurendeau. I thank him for the wonderful opportunity he gave me to work on such a motivating and inspiring project. He has helped me to develop multiple skills that will have an outstanding value for my future career.

I would like to thank Bombardier Aerospace, more specifically to the Advanced Aerodynamics department, for the great feedback throughout the entirety of this project. It would not have been possible without the financial support of Bombardier Aerospace, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the *Consortium de recherche et d'innovation en aérospatiale au Québec* (CRIAQ).

I would also like to acknowledge my research colleagues at Polytechnique Montréal. You have helped me learn so much, both in programming skills and scientific knowledge. Also, I would like to give special thanks to Simon, for his dedication to the success of our group and his well-advised knowledge. Our countless discussions have contributed to the success of this project, as well as making it the best work environment one could wish for.

My final thanks are dedicated to my family, for their support in all my adventures through all these years. Marlyne, Sylvain, Jasmine, thank you for always believing in me and helping me to become better in all aspects of my life. Alexandra, your presence at my side has been an incredible source of inspiration and joy.

RÉSUMÉ

Un des défis à relever pour les aérodynamiciens numériciens est de développer des méthodes représentant le plus fidèlement possible la dynamique des fluides. L'augmentation des ressources de calcul disponibles permet maintenant à la dynamique des fluides numérique de représenter et résoudre adéquatement ces problèmes. Les travaux présentés dans ce mémoire se concentrent sur le développement d'une méthode pour résoudre les équations de Navier-Stokes sur des géométries complexes. Le logiciel utilisé pour faire ces simulations est celui développé à Polytechnique Montréal, NSCODE. Deux objectifs sont définis pour le projet: développer une méthode permettant la résolution de géométries complexes utilisant des maillages partageant une surface et démontrer la robustesse de la méthode en lien à des applications de type industriel.

Dans le but d'augmenter les capacités de la méthode, une revue de littérature du développement de la méthode dans différents groupes de recherche, tels la NASA ou l'ONÉRA, a été faite. La méthode chimère, aussi connue sous son appellation anglaise «Overset», est choisie pour sa grande flexibilité à supporter des géométries complexes. Elle permet de mailler les différentes composantes d'une géométrie de façon indépendante entre celles-ci. Cela permet donc de simplifier la génération des maillages, étape complexe dans le processus de la dynamique des fluides numérique. La méthode chimère fait l'assemblage entre les différents maillages, utilisant des fonctions d'interpolation pour créer la communication entre eux. Une première version de la méthode avait précédemment été implémentée au sein du solveur NSCODE, mais n'avait été validée que sur des géométries dont les différentes composantes étaient entièrement entourées de fluide. Pour des géométries complexes, il n'est toutefois pas possible de procéder ainsi, et les maillages doivent pouvoir se superposer sur la surface de la géométrie.

Trois axes de développement permettant d'élargir les capacités de la méthode actuelle sont identifiés. Premièrement, la méthode telle qu'implémentée présentait un algorithme de découpe de géométries (traduction libre du terme anglais «hole cutting») sommaire, échouant sur des géométries concaves. Un algorithme utilisant une triangulation Delaunay contrainte pour modéliser la géométrie est venu renforcir cette étape de la méthode chimère. Deuxièmement, pour supporter des maillages qui se superposent sur la même géométrie, l'interpolation dans les régions visqueuses a été étudiée. Principalement, les particularités liées au solveur, soit une discrétisation centrée aux cellules et un schéma de dissipation artificielle requérant 2 voisins, sont venues influencer les choix pour la méthode. Le traitement adéquat de la condi-

tion frontière près de la surface ainsi qu'une limitation au processus de génération de maillage a permis d'obtenir une interpolation adéquate dans cette région complexe. Troisièmement, ces maillages doivent posséder un traitement spécifique pour calculer adéquatement les coefficients aérodynamiques. Une méthode de poids de panneau a été sélectionnée et implémentée, car elle a une influence minime sur la méthode de calcul des coefficients. Les méthodes implémentées ont été vérifiées en fonction de leurs différents aspects.

Finalement, des études de cas mettant en lumières les nouvelles capacités de la méthode chimère ont contribuées à mieux analyser la réussite de l'implémentation des différentes parties du travail. La partie la plus limitative de la présente recherche reste le traitement de l'interpolation dans la couche limite. Bien que la limitation à la génération du maillage produise un assemblage des maillages adéquat et que l'interpolation des propriétés soit juste, les propriétés plus sensibles aux gradients, tel le coefficient de cisaillement à la paroi, sont discontinues au travers de la frontière chimère. Cela montre le besoin de mener des études plus approfondies sur l'interpolation dans ces régions. Les autres aspects développés ont permis d'augmenter les capacités de la méthode chimère, ce qui montre des résultats intéressants pour simuler des géométries complexes.

ABSTRACT

Aerodynamics engineers aspire to develop methods that represent with as much fidelity as possible fluid dynamics. With the fast growth of computational resources, Computational Fluid Dynamics (CFD) tools can now rely on high fidelity methods to solve these problems. This thesis focuses on the development of a method to solve the Navier-Stokes equations over complex geometries. The flow solver developed at Polytechnique Montreal, NSCODE, is the software used to perform the simulations. Two objectives are defined: develop a method to simulate complex geometries using surface conforming meshes and demonstrate its robustness with respect to industrial type applications.

A literature review is conducted to evaluate the maturation of the overset method inside different research groups, notably the NASA and the ONERA. Also known as the Chimera method, it is selected based on its capacity to handle such difficult geometries. It allows to mesh different components individually, which ensures maximum grid quality. The mesh generation process is then simplified, which is regarded as a tedious and time consuming aspect of CFD. The overset method proceeds to perform the assembly of the different components together. Communication between these meshes is assured by using interpolation functions. An initial version of the overset method had previously been implemented inside NSCODE. Its validation was partially done, as it was only used for fully separated geometries. For complex geometries, this condition can not always be met, and the method must be able to treat meshes that overlap on the surface.

Three development axis are identified to increase the capabilities of the current implementation. First, the hole cutting algorithm in place, while being a fast method, lacks of versatility towards more complex cases. Concave geometries lead to non valid grid assembly. An algorithm is developed to replace it, which uses a constrained Delaunay triangulation to represent accurately the internal geometry. Second, in order to support meshes with overlapping surfaces, a study of the interpolation in the viscous region is performed. Focus is given to the particularities of the flow solver, mainly the cell centred scheme as well as an artificial dissipation scheme, to influence the chosen methods. Two aspects are analyzed: the mesh generation for these meshes and the proper treatment of the boundary condition. A limitation is proposed to the mesh generation, to help ensure adequate grid assemblies and valid interpolation donors. Third, the manner to compute the aerodynamic forces and moments is addressed. A weighted panel method is introduced to avoid the double integration in overlapping regions. It is chosen in light of its minimal impact on the computation of the

aerodynamic forces and moments. The implemented methods are verified with respect of their particular aspects.

Different case studies, showcasing the new capabilities of the overset method, are conducted to better analyze the success of this work. In light of these results, minor unresolved issues remain. Even with the adequate mesh limitation, gradient-based properties, such as the skin friction coefficient, is discontinuous at the overset interface. Such results highlight the need for a more profound analysis of the interpolation process in the case of viscous flows. Nevertheless, the other methods applied to the new overset method show its capability to simulate complex cases.

TABLE OF CONTENTS

ACKNOWLEDGMENTS III

RÉSUMÉ IV

ABSTRACT VI

TABLE OF CONTENTS VIII

LIST OF TABLES XI

LIST OF FIGURES XII

LIST OF ACRONYMS AND ABREVIATIONS XVI

LIST OF APPENDICES XVII

CHAPTER 1 INTRODUCTION 1
1.1 Basic concepts 1
1.1.1 Navier-Stokes equations 1
1.1.2 Spatial and temporal discretization 2
1.1.3 Mesh generation 2
1.1.4 Overset grid method 3
1.2 Definitions 3
1.3 Elements of the problematic 4
1.3.1 Complex geometries 4
1.3.2 Existing software 5
1.3.3 Requirements 5
1.4 Objectives 6
1.5 Outline of thesis 7

CHAPTER 2 LITERATURE REVIEW 8
2.1 Review of the overset method 8
2.1.1 Development of the method 8
2.2 Review of the method within the framework NSCODE 9
2.2.1 Flow solver 9
2.2.2 Overall presentation 11

2.2.3 Implementation and acceleration techniques . . . 13
2.3 Hole cutting . . . 18
2.3.1 Elimination process . . . 18
2.3.2 Constrained Delaunay triangulation . . . 18
2.3.3 Object X-rays method . . . 19
2.3.4 CAD geometries . . . 20
2.4 Solid-solid intersection . . . 20
2.4.1 Single surface definition . . . 22
2.4.2 Duplicated surfaces . . . 22
2.5 Aerodynamic forces and moments . . . 25
2.5.1 Hybrid mesh approach . . . 25
2.5.2 Weighted panel method . . . 26
2.6 Mesh generation in overset cases . . . 28
2.6.1 Automatic mesh generation . . . 28
2.6.2 Solid CAD meshes . . . 29
2.7 Usage of the method . . . 29
2.7.1 Multi-element airfoils . . . 30
2.7.2 Physical phenomena . . . 30
2.7.3 Unsteady simulations . . . 32

CHAPTER 3 IMPLEMENTATION AND VERIFICATION . . . 33
3.1 Modification of the preprocessor . . . 33
3.2 Hole cutting . . . 35
3.2.1 Internal mesh . . . 36
3.2.2 Blanked cells detection . . . 38
3.2.3 Verification . . . 40
3.3 Surface conforming meshes . . . 46
3.3.1 Effect of wall spacing . . . 46
3.3.2 Computational stencil correction . . . 46
3.3.3 Ghost cells definition . . . 47
3.3.4 Verification . . . 48
3.4 Computation of the aerodynamic loads . . . 53
3.4.1 Weighted Panel method . . . 54
3.4.2 Verification . . . 55

CHAPTER 4 NUMERICAL RESULTS AND APPLICATIONS . . . 59
4.1 Mesh generation . . . 59

4.2 Laminar flow . . . 59
4.2.1 Definition of the case . . . 59
4.2.2 Analysis . . . 60
4.3 Transonic flow . . . 62
4.3.1 Definition of the case . . . 62
4.3.2 Analysis . . . 63
4.4 Steady wing-aileron . . . 68
4.4.1 Definition of the case . . . 68
4.4.2 Analysis . . . 72

CHAPTER 5 CONCLUSION . . . 78
5.1 Synthesis of work . . . 78
5.1.1 Development . . . 78
5.1.2 Applications . . . 79
5.2 Limitations of the proposed solution . . . 80
5.2.1 Mesh generation . . . 80
5.2.2 Skin friction coefficient . . . 80
5.3 Future work . . . 81
5.3.1 Acceleration . . . 81
5.3.2 Grid analysis . . . 81
5.3.3 Automation of mesh generation . . . 82

REFERENCES . . . 83

APPENDICES . . . 92

LIST OF TABLES

Table 3.1	Parameters used for the verification of the hole cutting algorithm with the MDA case	42
Table 3.2	Comparison of lift and drag coefficients between the old and new overset preprocessor	42
Table 3.3	Parameters used for the verification of the boundary condition treatment with the cylinder case	49
Table 3.4	Parameters used for the verification of the boundary condition treatment with the NACA0012 case	51
Table 3.5	Grid convergence study for the verification of the weighted panel method for the surface length integration	58
Table 4.1	Drag comparison for the cylinder in laminar flow	60
Table 4.2	Parameters used for the transonic flow past the RAE2822 airfoil	63
Table 4.3	Comparison of the aerodynamic coefficients for the RAE2822 airfoil	64
Table 4.4	Parameters used for the NACA0012 airfoil in the wing-aileron configuration	69
Table 4.5	Mass flow rate in the gap region for the aileron deflection of 5.0°	73
Table 4.6	Aerodynamic coefficients for the three aileron configurations and five mesh techniques	77

LIST OF FIGURES

Figure 1.1	Example of a multi-block mesh with its main components	3
Figure 2.1	Computation stencil for the conservative and viscous fluxes (a), the artificial dissipation stencil (b) and the complete stencil (c)	10
Figure 2.2	Boundary conditions over a single block	11
Figure 2.3	Overset identification of the cells of the main element in a multi-element airfoil configuration	12
Figure 2.4	Recursive Virtual Cartesian Grid over an airfoil mesh (shown in green) © Pigeon, 2015. Reproduced with permission.	14
Figure 2.5	Original overset identification search tree algorithm © Pigeon, 2015. Reproduced with permission (translated from French).	16
Figure 2.6	Overset identification of the halo cells of the flap of a multi-element airfoil (only the first layer of halo cells is displayed)	17
Figure 2.7	Computed domain of a multi-element airfoil, showing in red the interface between the slat's computed cells and the main element's computed cells	17
Figure 2.8	Constrained Delaunay triangulation of a concave-convex shape, highlighting its initial constraints	19
Figure 2.9	Different types of overset configurations	21
Figure 2.10	Surface meshes of the DLR-F6 wing-body configuration with the wing mesh (red) and the fuselage mesh (teal)	21
Figure 2.11	Overset meshes for solid-solid intersections of multiple bodies	22
Figure 2.12	Surface discretization of overlapping meshes for a convex geometry (a-c) and a concave geometry (b-d), highlighting cells located outside of the computational domain (a-b) and the improper interpolation donors (c-d)	24
Figure 2.13	Hybrid mesh approach showing the blanked cells to remove overlap (a) and the triangulation added to cover the region (b)	26
Figure 2.14	Representation of the weighted panel method over an overlapping pair of surface cells	28
Figure 2.15	Space shuttle CFD result © Gomez, 2008	30
Figure 2.16	Multi-element airfoils, such as the McDonnell Douglas airfoil (MDA) (a) and the NLR7301 airfoil with its flap (b)	31

Figure 3.1 New overset identification search tree algorithm with the decoupled hole cutting 34
Figure 3.2 Concave-shape and cylinder test case studied to identify problematic hole cutting using the elimination process 35
Figure 3.3 Cross product over the starting indices of the block to determine its orientation 36
Figure 3.4 Advancing front method with the original nodes connected (a), the first triangle generated (b) and the final triangulation (c) 37
Figure 3.5 Comparison of vertex-centred (a) and face-centred coordinates (b) used in the hole cutting, with the face-centred coordinates used in the overset identification process 37
Figure 3.6 Hole cutting algorithm used to identify the blanked cells from the overset process 38
Figure 3.7 Representation of barycentric coordinates of a triangle 39
Figure 3.8 Constrained Delaunay triangulation for the concave-shape and cylinder geometry 41
Figure 3.9 Comparison of the overset grid assembly for the concave-shape and cylinder geometry 41
Figure 3.10 Hole cutting result for the MDA airfoil, with the old and new algorithm 43
Figure 3.11 Overset grid assembly for the MDA airfoil using the new overset preprocessor 44
Figure 3.12 Comparison of flow solution around the MDA airfoil for the old and new overset preprocessor 45
Figure 3.13 Comparison of convergence around the MDA airfoil for the old and new overset preprocessor 45
Figure 3.14 Comparison of low (a) and high (b) Reynolds number wall spacings with respect to relative error between the discretization of a cylinder by 1° arc lengths 47
Figure 3.15 Application of the computational stencil to a cell located at a corner of a mesh 48
Figure 3.16 Mesh generation process for a collar mesh with coincident surface discretization about the overset boundary condition 48
Figure 3.17 Meshes used for the verification of the boundary condition treatment 49
Figure 3.18 Cell-centre X-velocity about the patch boundary with and without computational stencil correction 50
Figure 3.19 Pressure (a) and skin friction (b) coefficients about the the patch border 51

Figure 3.20 Overset mesh over the NACA0012 airfoil for the verification of the ghost cells definition 51
Figure 3.21 Comparison of skin friction for the single grid case and overset grid when using linear extrapolation for the ghost cells 52
Figure 3.22 Pressure (a) and skin friction (b) coefficients for the proposed method in the overlapping region of the NACA0012 airfoil 53
Figure 3.23 Algorithm for the computation of the weight used in the aerodynamic loads computation 55
Figure 3.24 Overset grid assembly used for the verification of the weighted panel method 56
Figure 3.25 Illustration of the weighted panel algorithm result on the cylinder mesh 57
Figure 3.26 Grid convergence for the integrated surface length on a cylinder . . . 58
Figure 4.1 Streamlines and pressure contours past a cylinder for a single (a) and overset grid (b) 61
Figure 4.2 Convergence for the cylinder subject to a laminar flow for the single and overset grids 61
Figure 4.3 Pressure (a) and skin friction (b) coefficients comparison for the cylinder subject to laminar flow for the single and overset grids 62
Figure 4.4 Overset grid assembly of the RAE2822 airfoil 63
Figure 4.5 Convergence of the density residual for the single and overset grid configurations 65
Figure 4.6 Pressure contours of the RAE2822 airfoil for the single (a) and overset (b) grid solutions, at $Ma = 0.73$, $Re = 6.5 \times 10^6$ and $\alpha = 2.7225^{\circ}$ . . 66
Figure 4.7 Pressure (a) and skin friction (b) coefficients of the RAE2822 airfoil for the single and overset grid solutions, compared to experimental data 67
Figure 4.8 NACA0012 airfoil with the wing-aileron configuration 69
Figure 4.9 Dual C-type topology used for the small gap wing mesh 70
Figure 4.10 Dual C-type mesh generated around the main element with the poor quality cells in the aileron vicinity (a) and with the blanking applied (b) 70
Figure 4.11 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 5.0° 71
Figure 4.12 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 0.0° 74
Figure 4.13 Aerodynamics coefficients against the aileron deflection (δ_A) for the five mesh configurations 75

Figure 4.14 Pressure contours and streamlines in the gap region for the three gapped configurations 76
Figure A.1 Single grid used for the three aileron deflection angles 92
Figure A.2 Metrics of the dual C-type mesh generated for the small gap wing mesh in the aileron region 93
Figure A.3 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 5.0° 94
Figure A.4 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 10.0° 95
Figure A.5 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 0.0° 96
Figure A.6 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 10.0° 97

LIST OF ACRONYMS AND ABREVIATIONS

BC Boundary Condition.

CAD Computer-Aided Design.

CDT Constrained Delaunay triangulation.

CFD Computational Fluid Dynamics.

CFL Courant-Friedrichs-Lewy number.

FDM Finite Difference Method.

FEM Finite Element Method.

FSI Fluid-Structure Interaction.

FVM Finite Volume Method.

ISW Infinite Swept Wing.

MDA McDonnell-Douglas airfoil.

NLFD Non-Linear Frequency Domain.

OGP Overset Grid Preprocessor.

RANS Reynolds Averaged Navier-Stokes.

RCVG Recursive Cartesian Virtual Grid.

SSLV Space Shuttle Launch Vehicle.

URANS Unsteady Reynolds Averaged Navier-Stokes.

LIST OF APPENDICES

Appendix A Wing-aileron configuration 92

CHAPTER 1 INTRODUCTION

Aerospace engineering is in constant research for developing novel and more efficient means of flying through the sky. Over the last century, the study of fluid dynamics has helped to better comprehend the physics in action. At first, the only mean to obtain results was through experimental testing using wind tunnel models (Higgins, 1936). With the arrival of computers, numerical methods were developed to solve fluid dynamics problems. Computational Fluid Dynamics (CFD) regroups the numerical methods that allow the simulation of fluid dynamics. They are typically composed of four elements: a discretization scheme, a solver scheme, a support – namely a grid – and a post treatment.

These tools have enabled scientists and engineers to understand the behavior of fluids for more and more challenging cases. Over the last two decades, the computational power of high performance computers have increased by a magnitude of approximately 6 orders (Mavriplis, 2016). With this increase in resources available, engineers have been able to perform simulations using grids with hundreds of million points (Buning and Gomez, 2010). CFD is now a widely used tool to decrease the development cycle time of an aircraft and reduce the design risks in the aeronautics industry (Gregg, 2016). With the progression of numerical analysis to achieve certification of aircraft (Gregg, 2016), CFD must be performed on geometries that are as accurate as possible.

1.1 Basic concepts

This section briefly describes the general concepts underlying the resolution of CFD, from larger concepts, narrowing down to the subject of this thesis. It aims at locating the research performed within the CFD process. Specific concepts, such as turbulence modeling, will be discussed in their appropriate sections, rather than in this general discussion.

1.1.1 Navier-Stokes equations

Nowadays, CFD is applied to various simulation types, such as chemical reactions, internal combustions or internal flows. In the present thesis, external compressible flows are studied, which are governed by the Navier-Stokes equations which enforce conservation of mass, mass momentum and energy. The general form of this set of equations is presented as (Versteeg and Malalasekera, 2007):

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi\mathbf{u}) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \tag{1.1}$$

where ϕ denotes a property of the fluid (mass, momentum, energy), ρ is its density, t is the time, $\mathbf{u}$ is the velocity vector, Γ is the diffusion term and S_ϕ is the source term. By setting the ϕ value, with its corresponding diffusion and source terms, each equation can be retrieved. In order to solve these equations, spatial and temporal discretization must be performed.

1.1.2 Spatial and temporal discretization

The most largely used spatial discretization methods are: the Finite Element Method (FEM), the Finite Volume Method (FVM) and the Finite Difference Method (FDM). Most aerodynamic commercial CFD solvers, such as ANSYS FLUENT or Star-CCM+, use a FVM. In some FVM implementations, the properties are defined at the centre of the control volume and assumed constant throughout the control volume. This method uses an integral form of equation 1.1 over a control volume to solve the flow properties of interest:

$$\frac{\partial}{\partial t}(\int_{CV} \rho\phi dV) = -\int_A \mathbf{n}.(\rho\phi\mathbf{u})dA + \int_A \mathbf{n}.(\Gamma\nabla\phi)dA + \int_{CV} S_\phi dV \tag{1.2}$$

where CV represents the control volume and A is the bounding surface of the control volume. The left-hand side is the net rate of change of property ϕ. The right-hand side is often referred to as the residual and is comprised of respectively the convective, diffusive and source terms.

1.1.3 Mesh generation

The mesh generation process is of high importance in CFD, as it can take up to 50 % of the time spent on CFD simulations in the industry (Versteeg and Malalasekera, 2007). The spatial discretization of the domain is defined by vertices – also called nodes. These vertices are connected together to form cells and they compose the mesh. In many FVM formulations, it is the cell's centre that is of interest, because the properties are computed at this location.

Two categories of meshes exist: structured and unstructured. Structured meshes are composed of quadrilaterals – or hexahedra in 3D – with their neighbours known from the topological indices of the cell. In the topological space of structured meshes, the cells are identified by their indices, *(i,j,k)*, and the neighbours are preceding and following this indexation, with *+/- 1* position. Structured meshes can either be single or multi-block. A single block is a series of vertices that are ordered in a structured manner. Multi-blocks meshes are composed of multiple single block meshes that are connected together. Figure 1.1 highlights

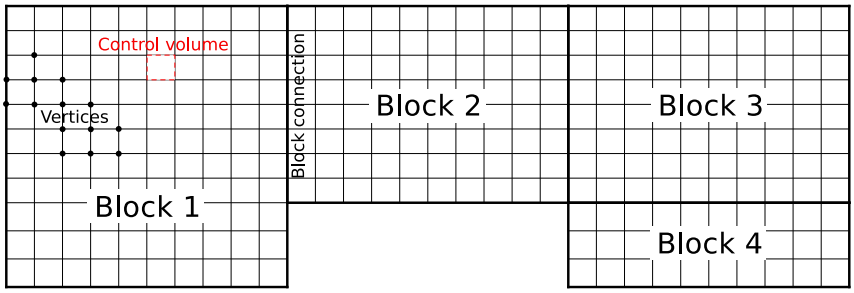


Figure 1.1 Example of a multi-block mesh with its main components

some of the main components of a multi-block mesh. Unstructured meshes are not ordered in the topological space as structured meshes are. They are typically composed of a larger variety of cells: triangles, quadrilaterals and more generic polygons – and their 3D equivalents. Thus, their neighbours are known by global indexing rather than topological indexing. This thesis focuses on structured meshes, as are used in many flow solvers: CFL3D (NASA), OVERFLOW (Boeing), Elsa (Onera), FANSC (Bombardier Aerospace) and NSCODE (Polytechnique Montréal).

1.1.4 Overset grid method

The Overset grid method – also known as the Chimera method – makes up the core of this thesis. This method uses multiple meshes to facilitate the treatment of complex geometries, such as multi-element airfoils in 2D or complete aircraft configurations in 3D. The general idea is that parts of the geometries are independently discretized by different meshes which superimpose one another. Information is transported from one mesh to the other using interpolation functions and thus, the flow solution can be obtained for the global case. CFD solvers that use this technique usually compute the information required, such as the overset identification and the interpolation weights, by a preprocessor, such as the one shown by Suhs *et al.* (2002). The preprocessor performs the overset grid connectivity, which assigns an identification to all cells and links the cells of the different meshes together. From that point, it is the flow solver that uses this information to compute the flow properties.

1.2 Definitions

Mesh terminology:

Grid	:	A system of meshes, forming the computational domain
Mesh	:	A set of blocks
Block	:	A set of vertices, arranged in a structured manner
Surface conforming mesh	:	also known as patch or collar mesh. It is a mesh that shares its surface with another mesh
Vertex	:	also known as node. The bounding points of a cell
Cell	:	The control volume on which the Navier-Stokes equations are solved (eqs. 1.1)

Overset grid terminology:

Hierarchy	:	Criterion used on a mesh to determine its relative dominance over other meshes
Overset interface	:	Region of the domain where overlapping meshes communicate, through interpolation, the primitive variables
Computed cell	:	Cell that is used for the computation of the flow simulation.
Interpolated cell	:	Cell that is interpolated from a donor cell.
Donor cell	:	Cell that is computed and whose primitive variables are used by an interpolated cell
Holecut cell	:	Cell that is located inside the geometry.
Orphan cell	:	Cell that do not possess valid donors.
Grid assembly	:	Refers to the set of cells that are computed for all the meshes in the case and that is used for flow visualization

1.3 Elements of the problematic

1.3.1 Complex geometries

The geometries that are handled nowadays by CFD solvers are not only larger (in grid sizes), but much more detailed than what was seen in previous decades. A good example of such an increase is shown on the Space Shuttle Launch Vehicle (SSLV), where the first studies simplified the surface by smoothing out sharp features (Buning and Gomez, 2010). On the recent configurations, the geometry is less altered, keeping most of the final components, such as the bipod ramp. This geometric fidelity is essential to obtain solutions that are coherent with experimental results. However, the more complex and detailed the geometries are, the higher the difficulty to generate good quality meshes. Simulations are typically performed on

aircraft composed of a wing, body, nacelle and tail, but might leave out flap track fairings, notches or other smaller components. These elements can cause flow perturbations that will not be captured by the CFD analysis.

As CFD is now a mandatory tool inside the development cycle of an aircraft and helps to both shorten the development cycle time and reduce the risks (Gregg, 2016), developing tools to handle complex geometries is of prime interest. The simulations must represent accurately the final product to minimize costs and increase the safety of aircraft. This can result in a reduction of required experimental testing and a diminution of unexpected behaviors during flight tests.

1.3.2 Existing software

Flow solver

One constraint of this project is that the solution developed must be compatible with the flow solver developed at Polytechnique Montréal, NSCODE. It solves the two-dimensional Reynolds Averaged Navier-Stokes (RANS) equations using multi-block structured meshes. The basis of the overset method was previously implemented inside the flow solver, as part of a preprocessor (Pigeon, 2015). While the most important features were included, such as the hole cutting and donor search, the implementation is not complete. There is no special treatment for intersecting geometries, which mitigates the utility of the method in its current form.

Grid assembly

In the overset grid method, a challenging problem is to validate the grid assembly after the overset grid connectivity is achieved. The challenge is to ensure that no orphan cell remains in the final grid assembly. Orphan cells are defined as cells that do not possess a valid donor. This can occur when the Overset Grid Preprocessor (OGP) does not find donors for a cell or when the donors are themselves identified as holecut or interpolated. Thus, an important step in treating with overset meshes is to analyze the grid assembly, look for these improper identifications and ensure that the overlap between the meshes is adequate.

1.3.3 Requirements

Automation

The original algorithm was developed with the objective to minimize user input.

The use of overset meshes represents a steep learning process for new users (Buning and Gomez, 2010), thus minimal user inputs eases the learning. In the case of the overset preprocessor developed, most cases can run without any user input, as the different parameters are set to lead to valid grid assemblies. The new developments that are to be included in the preprocessor must also respect this constraint. This ensures that new users can easily use the overset method and ultimately, this eases its technological insertion within industrial environment.

Efficiency

One of the main objective of the overset method is to increase the efficiency of RANS/URANS calculations. Hence, it is also an important consideration when developing the preprocessor. The resulting grid assembly must be able to maintain the efficiency of the flow solver with regards to convergence rate of the flow solution and wall-clock time to obtain the solution.

Robustness

As per its very nature, the overset preprocessor must be able to handle complex cases. Robustness comes in many forms, and in the case of the overset method, robustness can be defined as maintaining grid quality over the grid assembly. Grid quality refers to mesh metrics, such as orthogonality, skewness, aspect ratio and stretching ratio. Another aspect of the robustness is to successfully handle a vast range of configurations, such as meshes with shared surfaces.

1.4 Objectives

The work presented in this thesis aims at providing capabilities to solve CFD simulations over complex geometries. These include simulations with multiple bodies that can intersect. The mesh generation of these complex geometries must be kept in mind, as it influences the complexity that can be attained. The chosen approach derives from the project's constraint of using the solver NSCODE, which implies to work with multi-block structured meshes. From this main objective, the two specific objectives of this thesis can be defined as:

- Develop a method to simulate complex geometries with surface conforming meshes while respecting the project's constraints.
- Demonstrate the robustness of the developed approach and its readiness for industrial applications.

1.5 Outline of thesis

The thesis is separated into three sections: literature review, implementation and verification, and analysis via numerical results on various test cases.

The first section review the global literature on the overset method and the evolution of the method inside the research group at Polytechnique Montréal.

The second section presents the chosen methods and discusses their implementation inside the framework. These developments aim at providing a more robust way to handle complex geometries through the overset grid method.

The third section provides different applications that make use of the overset method and the developments presented in the second section. It aims at providing an overview of the capabilities of the overset method.

Finally, the conclusion presents a summary of the accomplished work as well as the limitations of the method. It also identifies future research directions.

CHAPTER 2 LITERATURE REVIEW

The literature review presented here focuses mostly on the developments of the overset method, with a special interest for some of the most advanced usage of the method. This fast growing method has seen its development rapidly integrated to many commercial software, such as Star CCM+ (Pinaev *et al.*, 2014) and the dedicated overset preprocessor Suggar++ (Noack *et al.*, 2009). The method is now also used within a large range of applications such as high-lift (Rogers *et al.*, 2000), rotorcrafts (Wissink, 2016; Coder *et al.*, 2016), wind farms (Mavriplis, 2016), fluid-stucture interaction (Miller *et al.*, 2014) or water simulations (English *et al.*, 2013). The general objectives of the overset method are to lower the complexity of the meshes generated, allow for simpler geometry motion and increase the complexity of the cases studied.

2.1 Review of the overset method

2.1.1 Development of the method

The ideas of the method are attributed to Joseph Steger, a reseacher at NASA (Steger *et al.*, 1983). In the mid 80s, a first implementation of the method was done with the PEGSUS preprocessor, coupled with the XMER3D flow solver (Benek *et al.*, 1986). It was first validated on an Euler-based implicit flow solver, and then expanded for viscous flows (Buning and Gomez, 2010). Since its first implementation, the overset method has been in constant development, mostly by NASA's research groups. At the time, NASA was already working with complex geometries, such as the space shuttle (Chiu, 1990), and this was one of the driving factor for the development of the overset grid method. With the arrival in the early 2000s of PEGASUS 5 (Suhs *et al.*, 2002) – the fifth version of the original overset preprocessor – the method had matured sufficiently to allow its usage for industrial applications. Since then, there has been a rapid growth of the developments made for this method, within NASA's groups and many other research groups.

Many OGP use the same overall procedure for the grid assembly (Deloze *et al.*, 2010; Chandar *et al.*, 2013; Roget and Sitaraman, 2014). It consists of two main components: an hole cutting process and an identification process. The hole cutting part consists of identifying the cells that are located inside of the geometries. These cells are particular, as they are located outside the computational domain. Hence, they do not participate in the computation of the simulation nor do they have flow properties. This process will be further discussed in section

3.2. The identification part is where the other cells – the ones inside the computational domain – will be identified as either donor or receiver. It is within that process that the weights of the donor cells are computed. In other words, it is this process that enables the communication between the meshes. This process will be further discussed in section 2.2, but a more detailed discussion is presented by Pigeon (2015).

2.2 Review of the method within the framework NSCODE

The framework NSCODE is used as a research platform for novel CFD techniques. It has been under development in the laboratory of Professor Eric Laurendeau at Polytechnique Montréal since 2012 and it has been validated through multiple studies (Bourgault-Côté and Laurendeau, 2015; Lévesque *et al.*, 2015; Mosahebi and Laurendeau, 2015; Parenteau *et al.*, 2015; Robitaille *et al.*, 2015). The framework presents both Euler and Navier-Stokes equations solvers, with explicit Runge-Kutta, point implicit and implicit LU-SGS schemes (Pigeon *et al.*, 2014). It is capable of solving both RANS and Unsteady Reynolds Averaged Navier-Stokes (URANS) equations, as well as featuring an ice accretion scheme (Bourgault-Côté, 2015) and a deformable structural scheme (Sansas, 2016). Some of its most novel methods include an Infinite Swept Wing (ISW) model (Bourgault-Côté *et al.*, 2017) and a Recursive Cartesian Virtual Grid (RCVG) method (Lévesque *et al.*, 2015). The OGP developed is built in a similar fashion than many other preprocessors (Pigeon, 2015).

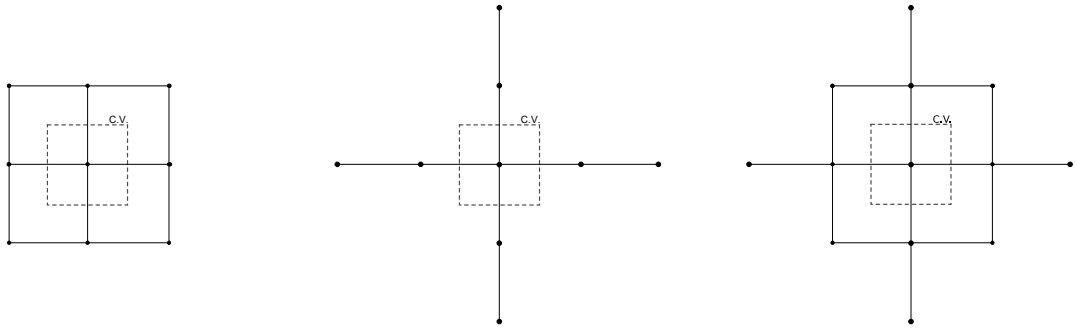
This section presents a brief description of the current implementation of the overset method within the framework NSCODE. The overset method depends on the architecture of the flow solver used, thus a description of the later is performed. Then, the overall presentation of the OGP is done and finally, the acceleration techniques used within the OGP are presented.

2.2.1 Flow solver

NSCODE solves the Navier-Stokes equations introduced previously in section 1.1.1 by using a FVM with a cell-centred scheme. In order to solve the equations, the fluxes on the control volume must be evaluated.

Figure 2.1(a) shows the cells that are required for the computation of the fluxes on the four faces of the control volume as well as the dissipation fluxes. NSCODE uses an artificial dissipation scheme which is implemented with two different formulations: a scalar dissipation scheme (Jameson *et al.*, 1981) and a matrix dissipation scheme (Swanson and Turkel, 1992). The usage of the artificial dissipation scheme requires an extension of the stencil shown in figure 2.1(a), because the direct neighbours of the cells need to have their direct neighbours

evaluated as well (figure 2.1(b). This brings the computation stencil to its complete form, as shown in figure 2.1(c).



(a) (b) (c)

Figure 2.1 Computation stencil for the conservative and viscous fluxes (a), the artificial dissipation stencil (b) and the complete stencil (c)

Boundary conditions

It can be seen from the stencil in figure 2.1(c) that each cell requires two direct neighbours in each direction, which is an issue at the borders of the blocks. To avoid this issue, NSCODE makes usage of a double layer of halo cells to ensure the proper computation of the fluxes. These halos are assigned a boundary condition type, which defines the equations to be used to compute their properties within these cells. This can be seen in figure 2.2, where each border is assigned its own boundary condition type. As NSCODE is a multi-block solver used for external flows, these boundary conditions are implemented:

- Non-slip wall condition;
- Slip wall condition;
- Far-field condition;
- Symmetry condition;
- Connecting condition;
- Overset condition.

Most of these boundary conditions are typical for CFD simulations. However, the "Overset condition" is not one typically available in CFD software as it is a condition specially

implemented for overset capabilities. This condition specifies that the halo cells must be interpolated, as this ensures that the cells located inside the domain have the knowledge from the overlapping mesh.

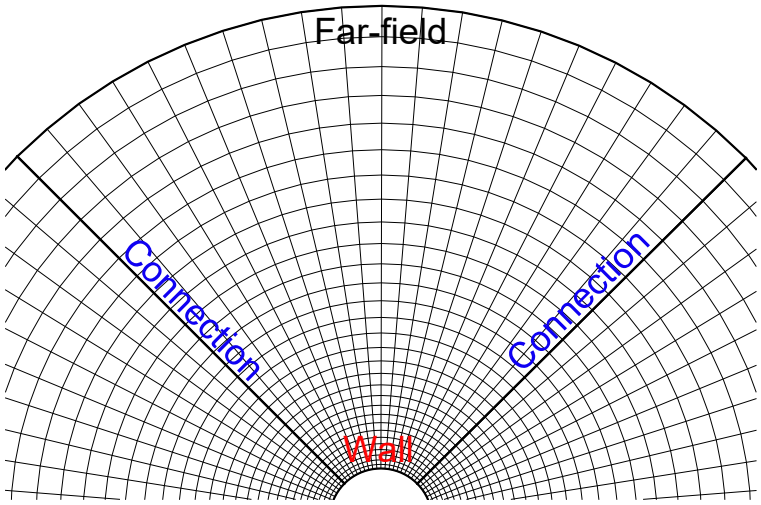


Figure 2.2 Boundary conditions over a single block

2.2.2 Overall presentation

This section details the structure and main features of the OGP. It also highlights some of the elements which will be part of future discussions within this thesis.

The main objectives of the OGP are to find the overset identification of the cells, determine the blanking for each cell and evaluate the weights used for the interpolation function of the interpolated cells. The overset identification of the cell defines its type, which can be:

- Computed cell;
- Dominant cell;
- Interpolated cell;
- Blanked cell;
- Buffer cell;

The computed cell refers to an identity where these cells do not have any overlapping cell from another mesh. The dominant cells refer to cells that are being overlapped by other cells from another mesh, but are dominant from these other cells. The interpolated cells refer to

these last cells, the ones that lost to the dominant cells. The blanked cells refer to cells that are located inside the geometry. The buffer cells are cells that are located at the interface of the dominant/interpolated cells and are treated as computed cells, to ensure that the domain is fully covered by computed cells. To further illustrate this identification, figure 2.3 shows all of the overset identification types of the main element of the McDonnell-Douglas airfoil (MDA). The slat and flap elements are shown with their wall boundary.

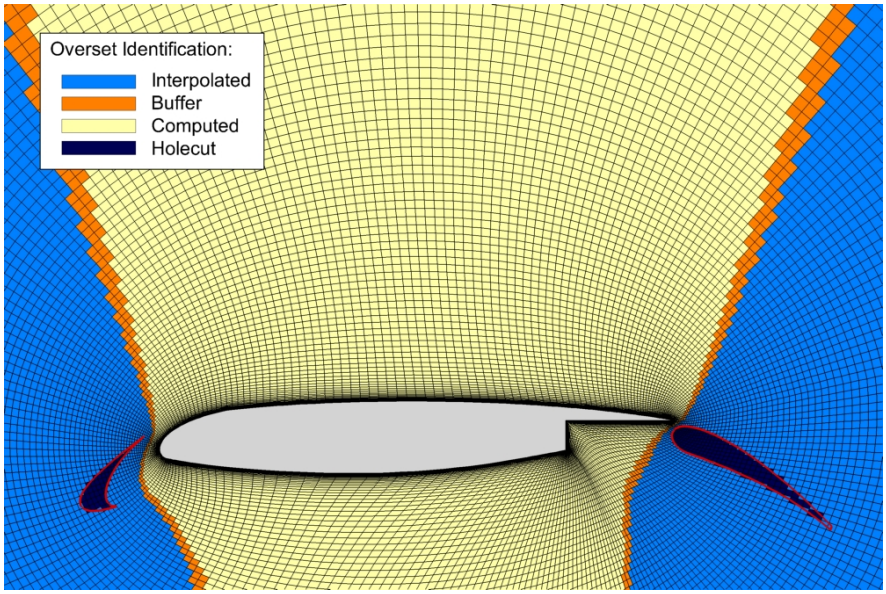


Figure 2.3 Overset identification of the cells of the main element in a multi-element airfoil configuration

The blanking separates into two categories the cells of the domain. On one side, there are the cells that participate in the solving of the flow, which are the *computed*, *dominant* and *buffer* cells. On the other side, there are the cells that do not influence the solving of the flow; they are comprised of the *interpolated* and *blanked* cells. For this category of cells, their residuals are put to zero, as to not influence the flow convergence. If the cell is an interpolated one, its properties – e.g. the density and velocity components – are updated with interpolation functions.

Finally, the interpolation weights of the interpolated cells are computed. Each interpolated cell has 4 donor cells which form a quadrilateral. The interpolated cell is located inside this quadrilateral and each donor cell possesses a weight, which is used in the computation of the interpolated properties of the cell.

2.2.3 Implementation and acceleration techniques

The OGP can be separated into multiple steps to account for the whole treatment:

1. Compute the metrics;
2. Create the RCVG, a spatial partitioning algorithm;
3. Evaluate the overset identity of each cell;
4. Find the donor cells of the overset boundaries;
5. Add the buffer zone between the computed and interpolated frontier;
6. Compute the weights associated with every interpolated cells;

Metrics

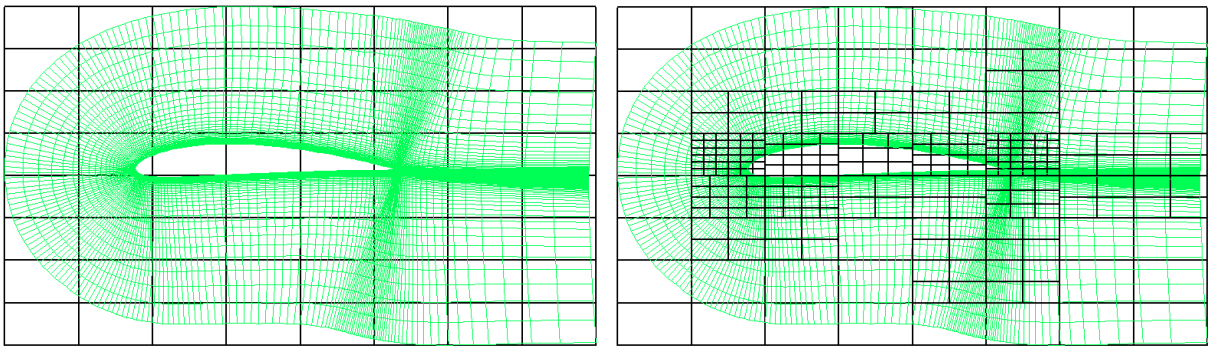
The metrics computed here are the ones that directly concern the OGP. Two metrics are used as criteria for the evaluation of the overset identity: the local wall distance and the cell volume (or area, as it is a two-dimensional solver). The cell area is a metric that is commonly used throughout the flow solver and it is already computed within the previous processes of NSCODE. The local wall distance represents the nearest distance from the cell to its own geometry. It is computed locally on the mesh of the cell, taken as if there were no other meshes present. In the current version of NSCODE, an Euclidean formula for the nearest wall node is used as the wall distance. The overset method uses either one of these two criteria, but it was shown that the cell area could lead to improper grid assemblies (Deloze, 2011). For this reason, it will not be part of further discussions, as the hierarchy combined with the local wall distance will be used throughout the rest of this thesis.

Recursive Cartesian Virtual Grid

The RCVG is a spatial partitioning method used in the overset identification to accelerate the search for valid donors. It consists of superimposing a virtual grid over the mesh, as shown in figure 2.4. The cells of the mesh are stored within their appropriate box – a cell from the Cartesian grid – which eases the search for superimposing cells. The method implements a novel approach were each box can be refined into a subsequent Cartesian grid. The subdivisions, N_x and N_y, are based on an average number of points per box:

$$N_x = N_y = \sqrt{\frac{N_{cells}}{\beta}} \tag{2.1}$$

where N_{cells} is the total number of cells of the mesh and β is the average number of points desired. This allows for a recursive number of refinements, were the original boxing, as shown in figure 2.4(a) can be refined by re-applying the same process inside each individual box that still have a high number of cells. It was shown that this method greatly accelerates the search in the overset identification part of the OGP (Pigeon, 2015).



(a) non-recursive grid (b) 2 levels recursive grid

Figure 2.4 Recursive Virtual Cartesian Grid over an airfoil mesh (shown in green) © Pigeon, 2015. Reproduced with permission.

Overset identification

This part of the OGP is one of the most important one, as it determines which cells are to be used and which are not part of the computed domain. In this step, the identity of each cell is determined based on the search tree shown in figure 2.5. It shows how each cell is treated according to its path through the search tree. The cells that go through the whole tree without passing through the red or green box are identified as computed. The ones that go through the green box are interpolated cells and finally, the ones that end up in the red box are the blanked cells. There are 2 criteria used to determine the overset identity of each cell: the hierarchy and the local wall distance. As mentioned previously, the cell area criterion is less reliable and will not be used in this study. The hierarchy is a user defined criterion and is applied to a whole mesh, whereas the local wall distance is proper to the cell and its superimposing cells. Because a cell can have a superposition with multiple other meshes, the algorithm loops through all the superpositions to ensure that the donor cells of an interpolated cell are the best suited based on the criteria. A superposition is defined as the cell-center of a cell being overlapped by cells from another mesh. It also ensures that all computed cells are the most dominant from all superimposing meshes. for example, a cell in mesh A could be computed from the point of view of a mesh B, but interpolated from the

point of view of a mesh C.

Overset Boundary Condition

As mentionned previously, the properties of the halo cells are not computed from the Navier-Stokes equations, but from a specific set of equations dependent on the boundary type of the cell. For the overset Boundary Condition (BC), the properties are obtained by interpolating from the superimposing meshes. Hence, every cell in the halos that have an overset BC type can only have an identification of "interpolated". The treatment to find their donors is similar to the process described in figure 2.5, but it is the cell in the boundaries that are tested with the other meshes' cells. Some halo cells can be located inside of another geometry, hence they are not properly computed. However, these situations occur far from the overset interface, as shown in figure 2.6, where the cells in dark blue are inside the geometry. For that reason, this issue does not influence the solution.

Buffer zone

The buffer zone serves to improve the passage of information from one mesh to the other. Figure 2.7(a) shows the computed domain – i.e. all the cells that are identified as computed – before the buffer zone is added. It can be seen that the interpolated cells located near the overset interface (shown in red in figure 2.7) can receive their properties from cells that are also identified as interpolated cells. This leads to an implicit interpolation, which is corrected by expanding the overlap between the meshes (Deloze, 2011). The cells near the frontier have their identification changed to computed and are further solved with the Navier-Stokes equations. Figure 2.7(b) shows the final computed domain with the buffer cells added.

Interpolated weights

Finally, every interpolated cells, which has been linked to its donor cells, need to have their weights computed and the interpolation structure must be populated. This is done by making a list of all the cells to be interpolated in the boundary updating process. The weights are computed by one of the interpolation function available in NSCODE, e.g. bi-linear or tetravolumic (Pigeon, 2015).

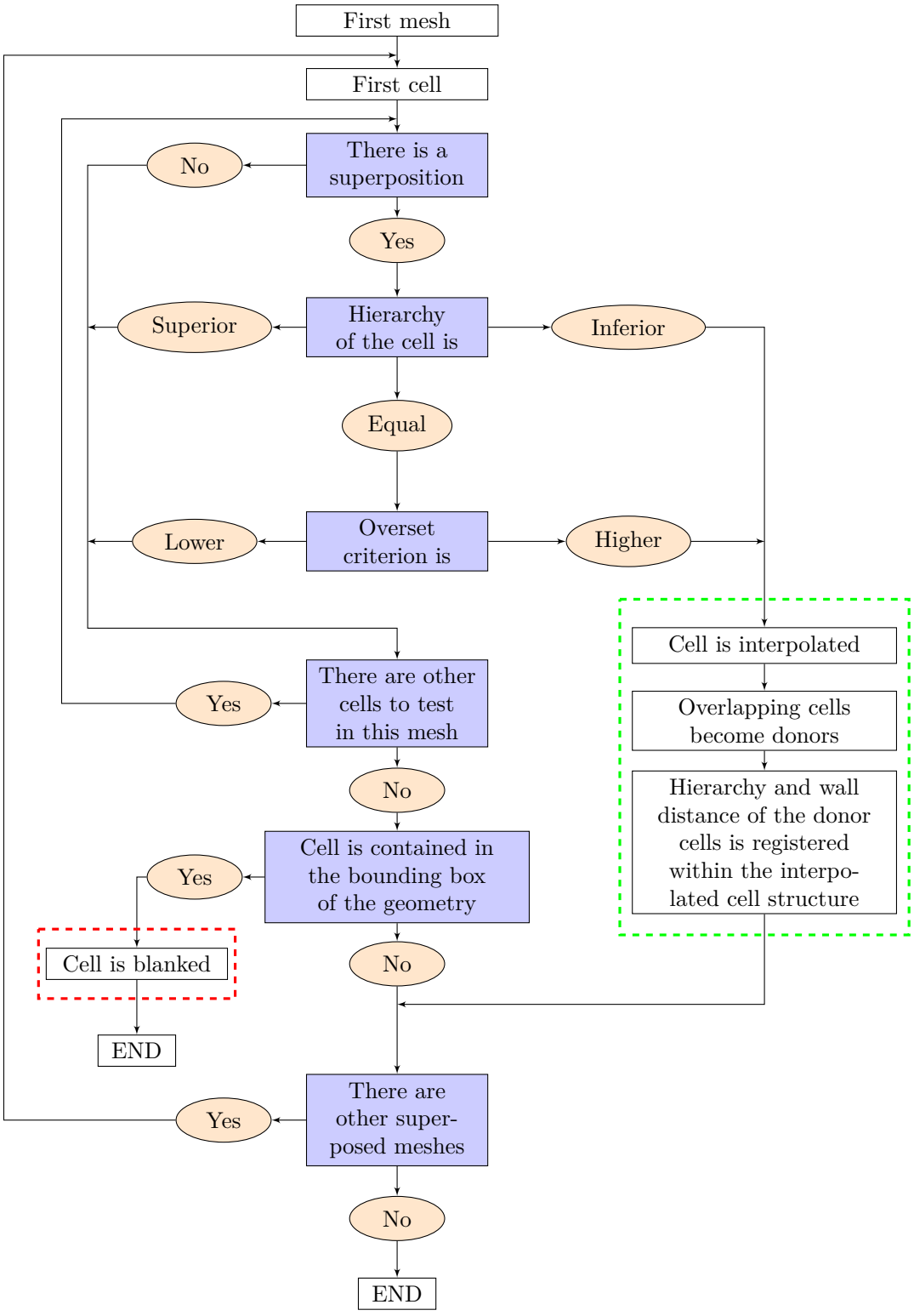


Figure 2.5 Original overset identification search tree algorithm © Pigeon, 2015. Reproduced with permission (translated from French).

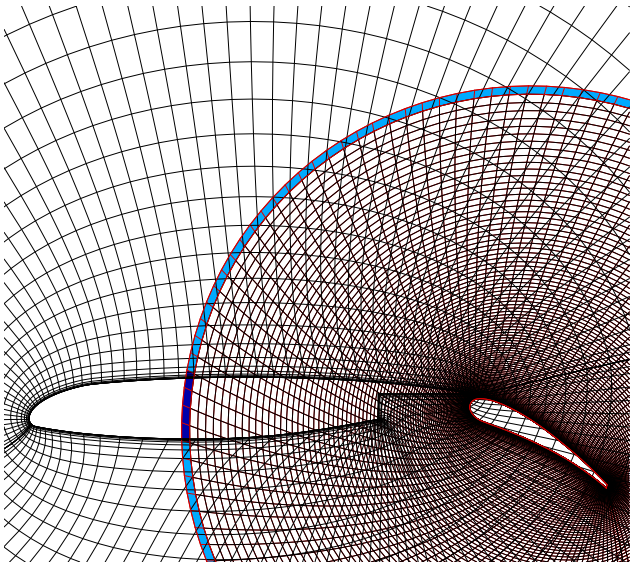
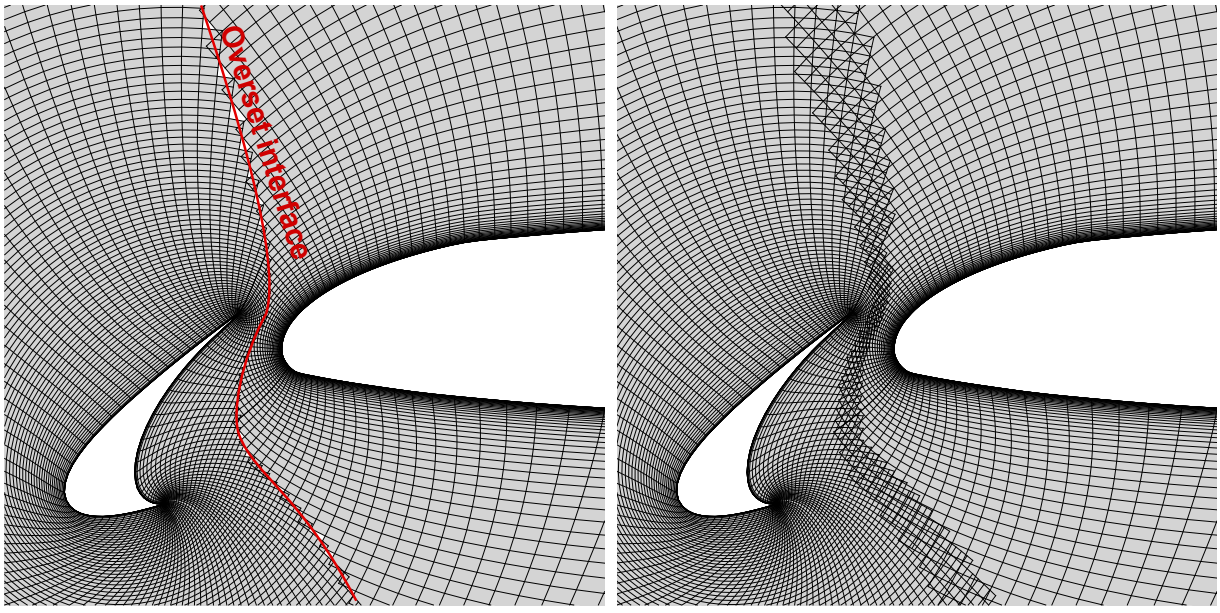


Figure 2.6 Overset identification of the halo cells of the flap of a multi-element airfoil (only the first layer of halo cells is displayed)



(a) No buffer cells (b) 2 layers of buffer cells

Figure 2.7 Computed domain of a multi-element airfoil, showing in red the interface between the slat's computed cells and the main element's computed cells

2.3 Hole cutting

This section aims at providing an overview of the different methods developed to identify the blanked cells. Hole cutting plays a major role in the overset method (Noack, 2007), because identifying the cells that are to be discarded helps reduce the computational cost of the simulation. Proper hole cutting is also of high interest, because it can directly influence the aerodynamic forces and moments (Chan *et al.*, 2013). Hole cutting methods can be separated into three different categories of algorithms: explicit, query and direct hole cutting (Noack *et al.*, 2009). Explicit methods require the most user input, and one of the objective of this thesis is to maintain a highly automated OGP. Therefore, this type of algorithms is not detailed in this literature review. A good comparison of the different types of algorithms, including strengths and weaknesses, is performed by Noack (2016). Novel direct hole cutting methods have been developed, with varying degree of complexity and execution time (Meakin, 2001; Kim and Chan, 2011; Chandar *et al.*, 2013). This section presents an overview of different methods, highlighting some of their strengths and weaknesses.

2.3.1 Elimination process

In the current version of NSCODE, the hole cutting is performed by an elimination process (Pigeon, 2015), based on the method from (Chandar *et al.*, 2013). It uses the assumption that if a cell is located inside the bounding box of a geometry, but did not find superimposed cells, it must then be located inside the geometry. This can be seen in figure 2.5, where cells that fail the other criteria are marked as blanked. This method offers the advantage to take a minimal execution time and has very little complexity. A simple test at the end of the overset identification is able to determine this status. However, due to its simplicity, it cannot handle complex geometries. In the case of concave geometries, this method can falsely identify cells as being blanked, when they should be computed, which is shown in section 3.2.

2.3.2 Constrained Delaunay triangulation

Constrained Delaunay triangulation (CDT) is used throughout a vast range of applications: geometric modelling, computer graphics rendering and FEM simulations (Dey, 2007). CDT is a subset of the Delaunay triangulation. This triangulation possesses unique features compared to other triangulations that are of high value for the applications at hand. The main characteristic of Delaunay triangulation is that the circumscribing circle of every triangles – the circle composed of the three nodes of the triangle – is empty of any other node. The

triangles are also optimized in relation to interpolation accuracy (Cheng *et al.*, 2012). The constraints added to the original Delaunay triangulation usually take the form of nodes that are already connected by edges which will be part of the final triangulation. By definition, CDT produces a set of triangles that resembles the most to a Delaunay triangulation, but respecting the constraints imposed to it. As for Delaunay triangulation, CDT can be refined to add more nodes to the set of original nodes forming the system. One of the first implementation of such a refinement method is attributed to Chew (1989). Figure 2.8 shows a simple CDT without any refinement, highlighting the initial constraints applied. It can be seen that complex shapes are well handled by the CDT. For the needs of hole cutting in an overset preprocessor, this feature is of high value, because the domain to mesh is predefined by the geometry nodes.

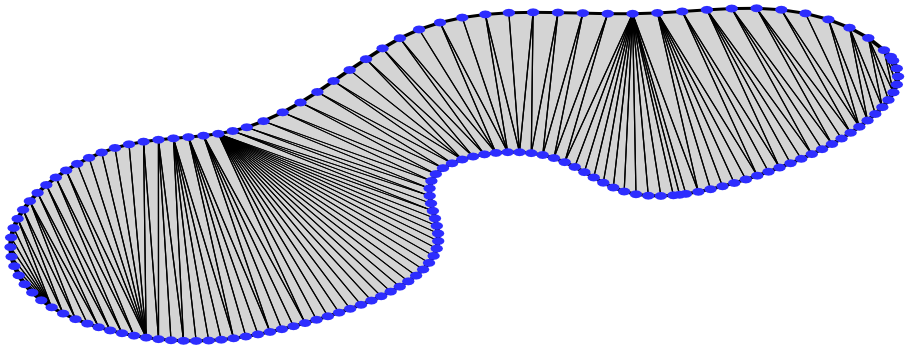


Figure 2.8 Constrained Delaunay triangulation of a concave-convex shape, highlighting its initial constraints

CDT have been used in multiple contexts for hole cutting: being it to close 3D geometries (Kim and Chan, 2011) or to ensure the passage of stencil-walk algorithms through geometries (Druyor *et al.*, 2015). For the application of hole cutting, this method consists of creating an exact mapping of the geometry with triangles in 2D – or tetrahedrons in 3D (Péron, 2016). In 2D, a CDT can be employed to fill the geometry, due to the great flexibility of the method to generate good quality triangular meshes and handle geometric curvature robustly (Togashi *et al.*, 2006a). One usage of the triangulation is to perform an overlap searching (Togashi *et al.*, 2006a). This allows to efficiently determine if a cell is located inside the geometry, by checking if it overlaps the CDT generated.

2.3.3 Object X-rays method

This method is one of the most complex method presented in this section. It transforms the geometries into objects to which a series of equally spaced "X-rays" in the XY plane (in 3D)

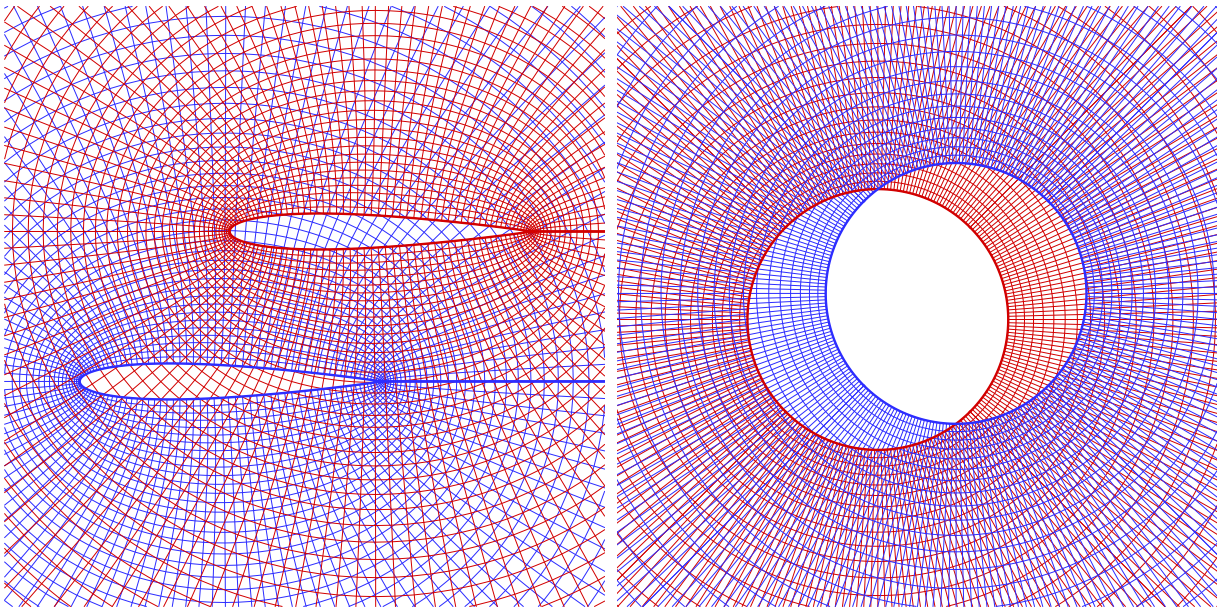
are projected to find the height at which the geometry begins and ends (Meakin, 2001). As an effort to reduce the complexity and user inputs, this method was extended to create a more automatic hole cutting process (Kim and Chan, 2011). This automated method uses Delaunay triangulation to seal the surface to a water tight surface. It then generates oriented bounding boxes of the components, to minimize the number of cells to be tested for blanking. The X-rays determine the height at which the geometry begins and ends inside this bounding box and it uses this information to produce the hole cutting for all cells inside the bounding boxes. This proves to be a robust method for complete aircraft configurations as well as geometries with very close components.

2.3.4 CAD geometries

The last hole cutting method presented consists of using Computer-Aided Design (CAD) geometries. This method uses the complete CAD to evaluate the points that are inside and outside of the geometry. Some novel development in this method uses a volumetric CAD generator (Haimes and Dannenhoffer, 2013) to be able to handle through a Boolean if the cell is inside the geometry (Dannenhoffer and Haimes, 2011). Because of the way this solid geometry modeller is built, the tests over the primitives are efficient, thus producing an hole cutting that is congruent to the geometry. In this particular case however, the shapes that can be generated are not yet to engineering level, as most of them are assembled from basic primitives, such as spheres, cylinders and such. Also, this technique only applies to 3D geometries, which is not suitable for the objectives of this thesis.

2.4 Solid-solid intersection

Another aspect that is not treated in NSCODE's overset preprocessor is the management of multiple bodies in contact. The current preprocessor was only validated with off-body simulations, such as multi-element airfoils or multiple airfoils, such as the one shown in figure 2.9(a). In off-body simulations, the different bodies – represented by a separated mesh – are completely surrounded by fluid, whereas solid-solid intersections aims at being able to perform simulation on multiple bodies in contact, shown in figure 2.9(b). Solid-solid intersections also aim at providing a mean for the mesh generation part to be able to separate a geometry into multiple overset meshes. For example, an aircraft could be separated into its main components, such as the fuselage, wing and nacelle. Literature shows many applications using these concepts such as the NASA's Space Shuttle Launcher Vehicule (Chan, 2009a), rockets configurations (Dannenhoffer and Haimes, 2011) or commercial aircraft (Hue *et al.*, 2015).



(a) Off-body geometries (b) Intersecting geometries

Figure 2.9 Different types of overset configurations

In order to treat these types of simulations, multiple methods have been developed throughout the literature. Most of these techniques rely on mesh generation of doubled surfaces. Doubled surfaces refer to multiple overlapping meshes based on the same original geometry. It can be seen in figure 2.10 that the wing mesh (in red) shares part of the fuselage surface (in teal). In that region, each mesh has its own surface discretization of the geometry.

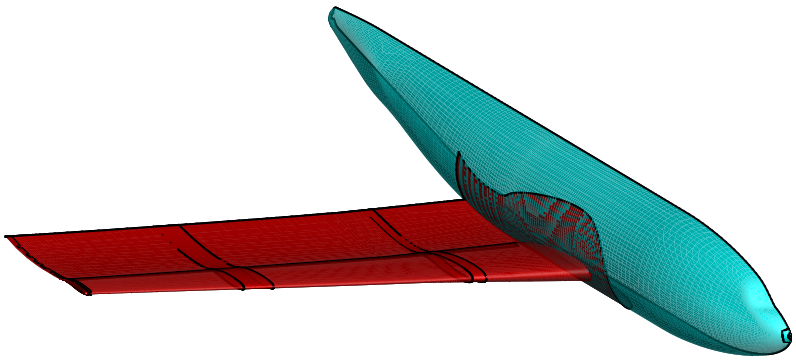


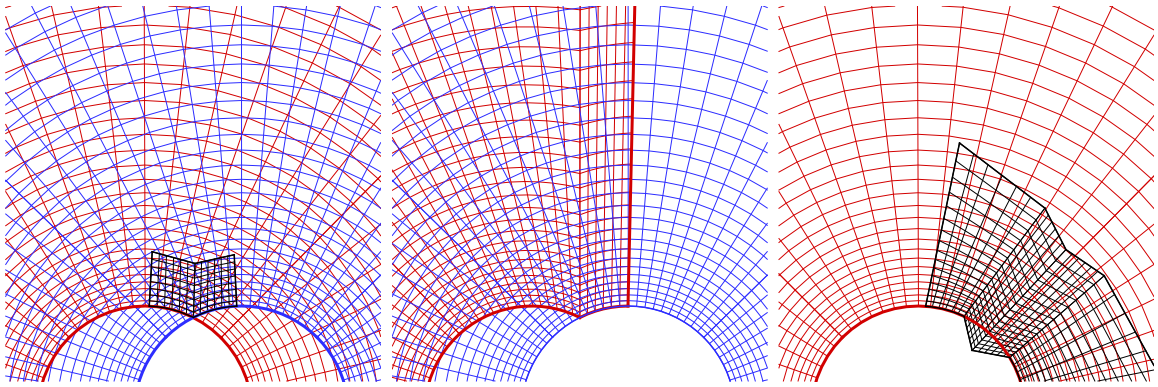
Figure 2.10 Surface meshes of the DLR-F6 wing-body configuration with the wing mesh (red) and the fuselage mesh (teal)

2.4.1 Single surface definition

Methods that do not imply an overlap of the surfaces are scarcely seen in the literature. One of the few implementations of such a method implies the usage of a grid-less solver to resolve the equations in the overlapping region (Benoit *et al.*, 2005). However, this method is limited to successful applications in the bi-dimensional range on non-viscous flows. To the author's knowledge, this method has not been thoroughly developed since then, showing limited applications for viscous simulations. Another method is implemented on a vertex based solver and consists of solving this issue on the edges of the mesh, to identify the solid intersections (Togashi *et al.*, 2006b).

2.4.2 Duplicated surfaces

For solid-solid intersections, most of the developments rely on duplicating parts of the geometries and then handling that configuration. Some developments make usage of collar meshes (figure 2.11(a)) which are located at the intersections between the geometries (Dannenhoffer and Haimes, 2011). Other developments use patch meshes (figure 2.11(c)) to add features – such as a spoiler over an airfoil or cavities – to the simulation (Blanc, 2010). Finally, one of the components can be meshed extended (figure 2.11(b)) to the connecting body (Hue *et al.*, 2015). These differences are highlighted in figure 2.11, where two cylinders are meshed with the different techniques. Altogether, these methods share the same idea: in order to succeed solid-solid intersections of multiple bodies, one of the mesh needs to have knowledge – i.e. a wall boundary – of the intersecting bodies A and B.



(a) Collar mesh, joining together the red an blue mesh

(b) Extended mesh, meshing the red mesh on the blue geometry

(c) Patch mesh, adding a cavity to the red geometry

Figure 2.11 Overset meshes for solid-solid intersections of multiple bodies

This method offers a relatively simple solution to this problematic, but multiple issues arise

from these kind of meshes. The major one is a discrepancy of the surface discretization between the two meshes. Although the meshes are discretized following the same original geometry, surface discretization can be different and thus, from a numerical point of view, the obtained geometry differs from one mesh to the other. This discrepancy can lead to two types of errors: improper interpolation donors or cells located outside of the computational domain (Suhs *et al.*, 2002). Figure 2.12 shows these differences for the two curvatures of geometries: convex and concave. It can be seen that in both cases, some cell centres fall outside of the computational domain – i.e. inside the geometry – from the point of view of the other mesh. This leads to a failure in the interpolation process, because these cells do not have proper donors to compute their properties. Figures 2.12(a) and 2.12(b) show the cells that have these kinds of issues. The next type of errors can be seen in figures 2.12(c) and 2.12(d), where an interpolated cell P, finds interpolation donors. However, the figures show that there can be a large difference between the wall distance of the local cell P – d_{local} – and its donor cells – d_{interp}. In RANS and URANS simulations, the properties' gradients are large near the wall, thus the interpolated properties from the donor cells are inaccurate (Suhs *et al.*, 2002; Schwarz *et al.*, 2010). This situation is particularly critical when resolving the Navier-Stokes equations with high Reynolds numbers, because the cells' height at the surface needs to be small. This leads to an higher discrepancy between the different meshes and interpolation errors are larger. In figure 2.12, the cells shown are relatively large – to better show the issue discussed and for clarity – but with high Reynolds simulations, the errors cover an increased range of cells. This leads to incorrect boundary layer profiles, which eventually leads to an improper flow solution.

To correct these errors, multiple methods have been developed in the literature to ensure valid interpolation donors in the near wall regions. In NASA's PEGASUS 5 overset preprocessor, the interpolation is corrected by transforming the coordinates of the receiving mesh to fit over the donor mesh (Suhs *et al.*, 2002). Then, with this new mesh, the interpolation donors and coefficients are computed. For the flow resolution, the original – untransformed – meshes are kept, but with the donor information gathered on the transformed meshes. This method has been implemented and integrated into the overset preprocessor used by the NASA and multiple research groups. It has been validated through multiple studies (Buning and Gomez, 2010; Chan, 2009a).

A newer method consists of computing the projected coordinate of an interpolated cell (Schwarz *et al.*, 2010). To compute the projection of the interpolated cell P, the method is divided into three steps. First, its nearest wall face is found, which corresponds to the local wall distance of the cell. Then, this wall face is projected onto the overlapping mesh's wall faces. This gives the relative error between the local wall distance and the interpolation wall

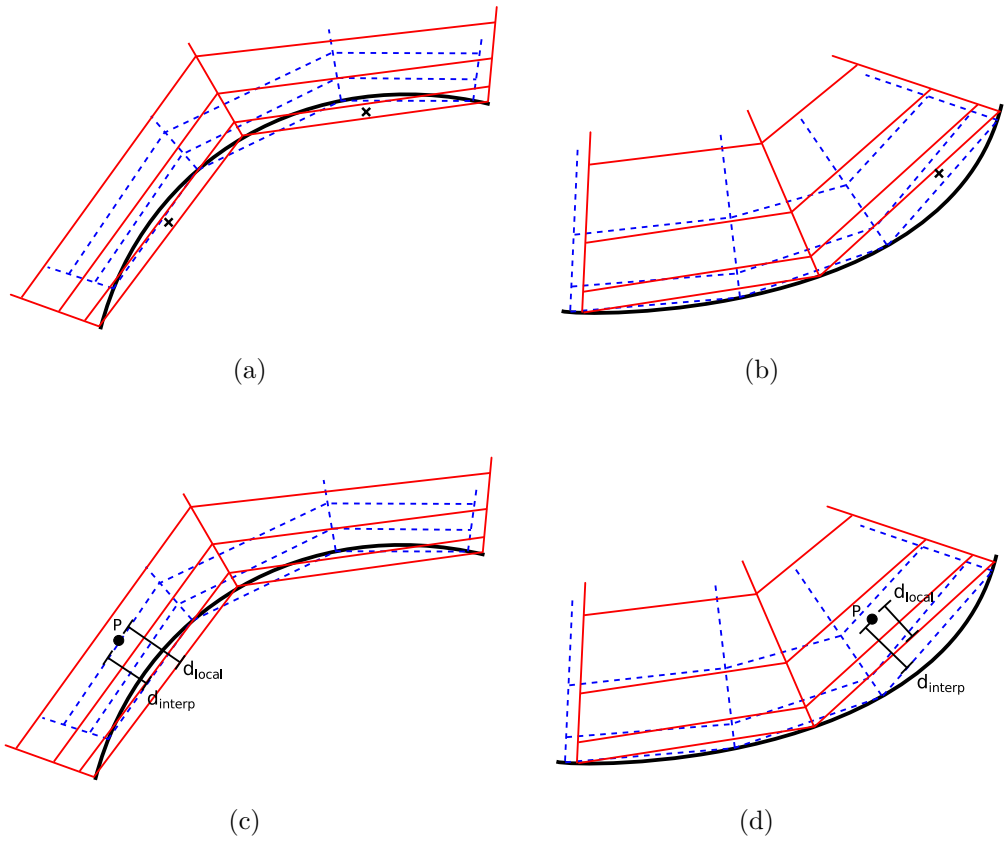


Figure 2.12 Surface discretization of overlapping meshes for a convex geometry (a-c) and a concave geometry (b-d), highlighting cells located outside of the computational domain (a-b) and the improper interpolation donors (c-d)

distance. Finally, the coordinates of the cell are adjusted based on equation 2.2:

$$\mathbf{x_{P^*}} = w\boldsymbol{\epsilon} + \mathbf{x_P} \tag{2.2}$$

where $\mathbf{x_{P^*}}$ is the projected coordinates of the cell, $\mathbf{x_P}$ is the original coordinates and $\boldsymbol{\epsilon}$ denotes the projected error between the two meshes. The term w is a weight added to control where the correction is applied, because corrections should only be applied in the boundary layer region. The weight takes values varying from 0 to 1: 0 is outside of the boundary layer and 1 is inside. To ensure a proper transition, the weight is gradually decreased near the edge of the boundary layer. This method proves to be an efficient method to adjust the coordinates of the cells that require modifications, while retaining the generated mesh intact. It has been validated both on two and three dimensional cases (Schwarz *et al.*, 2010).

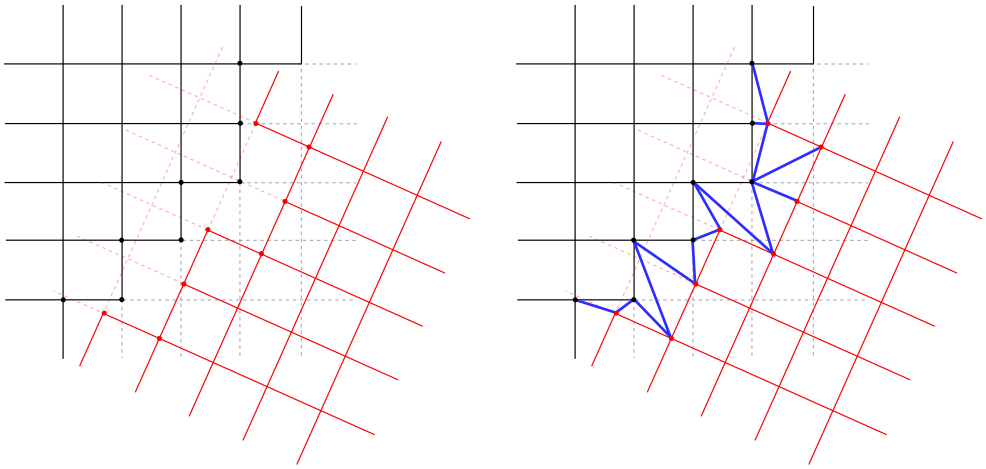
2.5 Aerodynamic forces and moments

Another challenge with solid-solid intersections arises in the computation of aerodynamic loads. These loads, which include the lift, drag and pitching moment (in 2D), are important metrics provided by CFD analysis. The aerodynamic loads are computed by an integration of the properties on the surface. Because the surfaces are doubled, the integration of the aerodynamic loads will compute parts of the geometries twice as well. This leads to incorrect forces and moments, which is known to be one of the flaws of the overset method (Buning and Gomez, 2010). This issue has been studied and multiple methods have been developed to address it. The first method developed relies on an hybrid mesh approach to ensure that the surface is completely covered by the integration. The second method, which consists of applying a weight coefficient to each integrated surface, was developed to address some issues from the first method, namely the robustness and the computing time required for the method. Some recent development on the hybrid mesh approach was able to correct these issues, and thus, offers two valid methods to compute the aerodynamic loads. Analysis of the two methods show that generally, the hybrid mesh approach produces more accurate results compared to the weighted panel method (Chan, 2009b).

2.5.1 Hybrid mesh approach

The general concept of the hybrid mesh approach consists of recreating a new surface mesh which does not possess any overlapped cells (Chan and Buning, 1995). To achieve that, all blanked cells are removed from the surface domain and all overlapping cells are removed to make a clear path between the meshes. It is shown in figure 2.13(a), where the nodes to be connected are highlighted by the dots. Then, the algorithm automatically re-connects the frontier nodes to one another, to produce a water tight surface, as shown in figure 2.13(b). The aerodynamic loads are integrated based on this new surface mesh, rather than the original overlapping meshes. The new mesh is composed from the non-overlapping quadrilaterals as well as the triangles used to "zip" the overlapping meshes together, to which this method owns its first appelation of "Zipper grids" (Chan and Buning, 1995).

In the original algorithm (Chan and Buning, 1995), weaknesses showed up when increasing the complexity of the overlapping meshes. The main weakness was a lack of robustness when having overlapping meshes with large discrepancies in the surface discretization. This lead to an increase of the user input required to produce a valid hybrid mesh. Also, with the increase of the number of cells and number of meshes, the search algorithm proved to lack efficiency which resulted in a large computational time for the algorithm. Some recent developments



(a) (b)

Figure 2.13 Hybrid mesh approach showing the blanked cells to remove overlap (a) and the triangulation added to cover the region (b)

were conducted on the hybrid mesh method to fix these weaknesses (Chan, 2009b). The main new features are the usage of a tree-based method – for the bounding boxes of the meshes – and the storage of the interpolation stencils rather than the computation when needed. The other main part of the new developments is an improved triangle stitching by splitting the frontiers of each mesh into separated strings, where each string is composed of nodes that will all connect to the same frontier. Then, the strings are linked together in pairs which will be connected between each others by the triangles. Overall, the algorithm was completely rewritten and possesses a speed-up of about 40 compared to the previous algorithm (Chan, 2009b), while producing good quality surface meshes with less user input.

2.5.2 Weighted panel method

The weighted panel method was developed in the early 2000's to address some of the issues that were present in the old version of the hybrid mesh method (Wigton, 2004). The current method comes from the software Unique Surfaces Using Ranked Polygons – USURP – (Boger, 2006) which is based on the work from L. Wigton (Boger and Dreyer, 2006). This algorithm, as well as the hybrid mesh method, are available within NASA's Chimera Grid Tools software package (Chan and Pandya, 2014). In opposition to the hybrid mesh method, where the integration of the forces occurs on a newly created surface mesh composed of quadrilaterals and triangles, the weighted panel method integrates the properties of all the surfaces' quadrilaterals. To ensure that the surface is not integrated twice, each surface cell

is assigned a weight which ponders its contribution to the global solution. The weight is a float value between 0 and 1, based on a polygon clipping algorithm.

The first step of the algorithm is to find overlapping pairs of elements. These overlapping pairs are found by performing a spatial search method using a R-tree (Boger and Dreyer, 2006). Once the overlapping pairs have been determined as valid, the rank of each surface element is evaluated. The rank is first determined by looking at the overset identity of the cell. Then, for pairs with the same overset identity, two different methods can be used: a "panel-based" or a "patch-based" (Boger and Dreyer, 2006). The rank of the surface cells is then used to compute a polygon Boolean difference on each pair of overlapping cell. This Boolean operator removes parts of the surface that is overlapped by an higher ranked cell. Finally, the weight coefficient is computed by summing the areas for each cell and performing a ratio with the final and original areas. Figure 2.14 highlights this process, by showing an overlapping pair. In this particular case, the respective weights can be expressed as:

$$w_1 = 1.0$$
$$w_2 = \frac{A_2 - A_{OV}}{A_2}$$

In a more general case, the overlapping area, A_{OV} is expressed as:

$$A_{OV} = \sum_{i=0}^{N} A_{OVi}$$

where A_{OVi} represents the overlapping area of each overlapping pair that includes the cell labelled as "A_2" and can be positive or negative, depending on the rank of the cell in the pair.

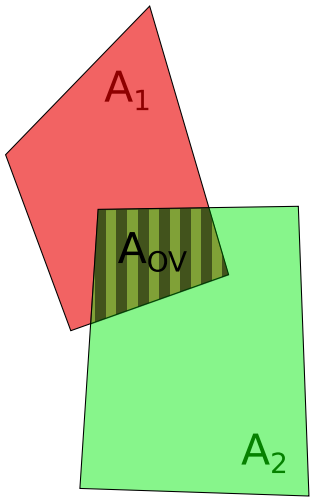


Figure 2.14 Representation of the weighted panel method over an overlapping pair of surface cells

2.6 Mesh generation in overset cases

Mesh generation is the process that is the most affected by the usage of the overset method. Effectively, the main objective of the overset method is to allow simpler mesh generation, both for unstructured and structured meshes. When using overset meshes, a common practice is to separate a case study into it components. For example, in an aircraft configuration, the main components would be meshed separately, such as the fuselage, the wings, the ailerons, the tail and also the domain. This method also allows to include or discard smaller components, such as fairings and winglets. Many papers address mesh generation for overset cases (Chan *et al.*, 2002; Pandya *et al.*, 2005). As was previously mentioned, mesh generation over complex geometries can take up to 50 % of the time for the global analysis of the case. Hence, it is a largely targeted area to decrease the costs of performing CFD simulations. Nowadays, dedicated software exist to ease the generation of overset meshes, such as Pointwise (Wyman, 2014).

2.6.1 Automatic mesh generation

A special attention is also taken to automatic mesh generation. Because the overset meshes represent less complex geometries, it is possible to automatically generate good quality overset meshes with minimal user input. One of the first automatic overset mesh generation method seen in the literature is an adaptive mesh refinement (Matsuno *et al.*, 1998). In this

method, overset meshes are generated near high-gradient flow properties and are solution-refined to better capture the physics. In the overset method, a typical meshing practice consists of separating the near-body mesh – a curvilinear mesh that conforms to the surface – and the off-body mesh – the one that represents the computational domain. For the off-body meshes, simple Cartesian meshes can be used, and thus, the automation process is not too difficult. One method is to adapt a coarse Cartesian grid, which defines the domain boundaries, to the flow properties gradient by adding a finer Cartesian grid layer (Katz *et al.*, 2009). Some more complete implementation of the automatic mesh generation also include the near-body automatic mesh generation (Benoit and Péron, 2012). In this implementation, the near-body meshes are generated to fit the boundary layer and the rest of the domain is covered by Cartesian meshes. Automated mesh generation has also been developed to independently mesh the main component and its smaller components (Pandya *et al.*, 2009). In this method, which is oriented towards axi-symmetric geometries, the axi-symmetric components are meshed together and the non axi-symmetric components are meshed based on collar meshes. More recent developments of fully automatic capabilities for surface mesh generation are being pursued (Chan, 2016).

2.6.2 Solid CAD meshes

Another area of research is to change the surface CAD representation of the geometry to a volumetric CAD geometry. In this method, the geometry is represented by a parametric representation. The software Engineering Sketch Pad (Haimes and Dannenhoffer, 2013) employs the strengths of volumetric CAD to accurately compute the Boolean intersection of the different components and operations, such as unions, extrusions and chamfers. The software structures the components into a feature tree, which can be used to generate an overset mesh of the assembled geometry (Dannenhoffer and Haimes, 2011). The method developed is closely linked to the development of the CAD software and is highly automated. First, the surface meshes of the primitives are generated. Then, collar grids associated with the Boolean operations and the feature tree primitives are generated. Finally, the domain mesh is generated, with a Cartesian mesh. This produces a set of meshes that efficiently represents the case studied, because the collar grids overlap all of the components properly and the discontinued elements – the edges – are meshed separately from the surface meshes.

2.7 Usage of the method

The overset grid method is used by research groups, industries and CFD specialists from all around the world (Blanc, 2010; Togashi *et al.*, 2006a; Schwarz *et al.*, 2010; Rogers *et al.*,

2000; Benoit *et al.*, 2005; Sclafani *et al.*, 2008). The following sub-sections highlight this great flexibility by presenting studies done from multiple research laboratories. It focuses mainly on the cases studied and their results, rather than the methods used to obtain them, as they have for the majority been covered in the previous sections.

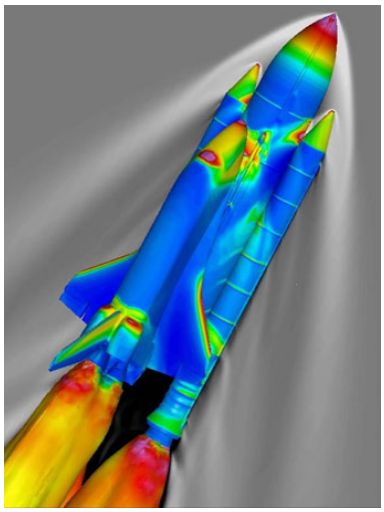


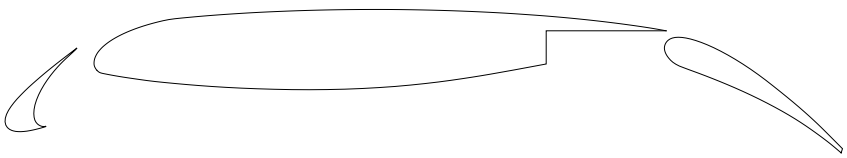
Figure 2.15 Space shuttle CFD result © Gomez, 2008

2.7.1 Multi-element airfoils

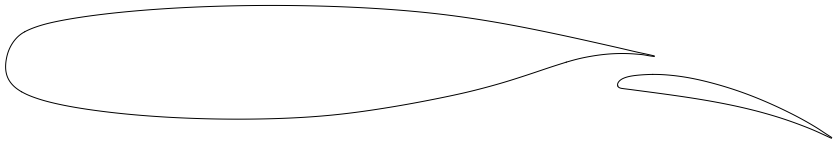
The overset method is particularly efficient with simulations were the bodies are completely separated from one another. This can be seen with multi-element airfoils, such as airfoil-slat-flap configurations (figure 2.16(a)) or airfoil-flap configurations (figure 2.16(b)). It is thus easy to move the meshes with relative motion between one another, because the mesh from one body is independent from the other body. Many studies have been conducted on these configurations. Liao *et al.* (2007) performed a study comparing the experimental and numerical results of the overset method on two cases: the NLF7301 airfoil with a flap and the three elements MDA. The numerical pressure coefficients distribution showed great agreement with the experimental values for both cases and highlights the capabilities of the overset method. Lévesque (2015) performed an optimization of the lift coefficient for the flap position on the 30P30N configuration of the MDA.

2.7.2 Physical phenomena

Another area of high interest for overset meshes is to accurately capture physical phenomena. One of these phenomena is the transonic shock over an airfoil (Kultajev *et al.*, 2011; Matsuno



(a)



(b)

Figure 2.16 Multi-element airfoils, such as the McDonnell Douglas airfoil (MDA) (a) and the NLR7301 airfoil with its flap (b)

et al., 1998). In structured mesh generation, it is possible to locally refine a small portion of the domain, but it is difficult to de-refine this zone. This often leads to a propagation of the refinement, which can lead to slower convergence and an increase of the total number of cells. Overset meshes offer an easier way to refine at the location required and stop this cell density propagation within the intended region. For example, the authors cited here studied the RAE2822 airfoil in conditions where a supersonic shock occurs on the upper surface. They were able to obtain an high resolution shock with coarser airfoil mesh, by superimposing a small located overset mesh.

Another interesting physical phenomena studied is Von Karman vortices. (Lévesque, 2015) studied these vortices over a cylinder whose mesh is rather easy to represent. In order to be able to keep a good mesh refinement in the wake region, a refined overset Cartesian grid was used. The results showed that this was an easy way to allow the Von Karman structure to propagate and dissipate farther away in the domain compared to the single grid.

The sonic boom of a rocket was simulated using both unstructured and structured meshes with the overset method (Ishikawa *et al.*, 2010). The unstructured mesh was used to capture the details of the rocket configuration, whereas the structured mesh was used for its efficiency in the axi-symmetric domain around the rocket.

2.7.3 Unsteady simulations

Multiple studies have been conducted on unsteady simulations using the strength of the overset method to avoid re-meshing, or at least to ease re-meshing. A Fluid-Structure Interaction (FSI) simulation was performed on a deforming beam attached to a cylinder which showed that the usage of overset meshes helped to increase the mesh quality of the simulation (Miller *et al.*, 2014). The results also showed that splitting the domain using overset meshes decreased the computational cost of the mesh movement algorithm. Typical FSI simulations can be limited by the large curvature of the geometries, when they are subject to large deformations. This has been highlighted as a problematic, particularly when studying the flutter condition of flexible beams (Sansas, 2016).

The study of unsteady flap movement is also greatly eased by the usage of overset meshes. For example, it allows to study a simulation where both the airfoil and the flap are subject to independent deflections (Liggett and Smith, 2013). In this configuration, each element – the airfoil and the flap – can be moved without having to be re-meshed.

Other unsteady simulations include store separation simulations, which is also a typical validation case for the overset method (Wang and Parthasarathy, 2000; Zhang *et al.*, 2015; Xuefei *et al.*, 2015). The usefulness of the method is clearly highlighted in these articles by allowing the body to move away from the wing. In general, the overset method has been used over many types of moving bodies, such as the SSLV, boosters separation (Gea and Vicker, 2006) or a missile head separation (Jingjing and Chao, 2010).

The aim of the literature review was to showcase advanced developments and cases studied with the overset mesh. It was shown that this method has gained interest in the last decade, and that its usage is still growing. The goal was also to highlight some of the novel methods developed to increase the robustness of the overset method, while putting it in relation with the OGP currrently available in NSCODE. Its strengths and weaknesses were described and places the context to discuss further improvements to the overset method.

CHAPTER 3 IMPLEMENTATION AND VERIFICATION

This chapter will present the holes that have been identified in the current implementation of the overset method and the chosen methods to correct them. These include a revamping of the hole cuttting process, the treatment of surface conforming meshes and a method to accurately compute the aerodynamic loads with cases using multiple meshes that share a common surface.

3.1 Modification of the preprocessor

To include the developments presented in the following sections, the preprocessor requires some modifications; the main one being the decoupling of the overset identification and the hole cutting. The overset identification process received a slight modification, which is highlighted in figure 3.1. It can be seen that in the previous algorithm (figure 2.5), the hole cutting was integrated inside the overset identification, whereas in the new algorithm, the hole cutting is done beforehand and these cells are not included in the overset identification. This allows to use a dedicated hole cutting algorithm and to be able to ease its implementation.

The next subject does not concern a real modification rather than an update of a parameter's definition. In the previous implementation, the hierarchy was introduced as a criterion for the overset identification (Pigeon, 2015). The hierarchy states the relative dominance of one mesh over the others. It is applied to a whole mesh, rather than cell by cell, and overrules the other criteria applied. In fully separated geometries, the local wall distance is enough to obtain a valid grid assembly, which makes seldom use of the hierarchy. This criterion, given by the user, gets much more meaning when using surface conforming meshes. In these cases, the local wall distance is the same on each mesh, because they share the geometry. Thus, grid assembly with this criterion fails. The hierarchy is here re-introduced: a surface conforming mesh must have an higher hierarchy than its overlapping mesh. This ensures an adequate grid assembly for these types of meshes, particularly near the wall.

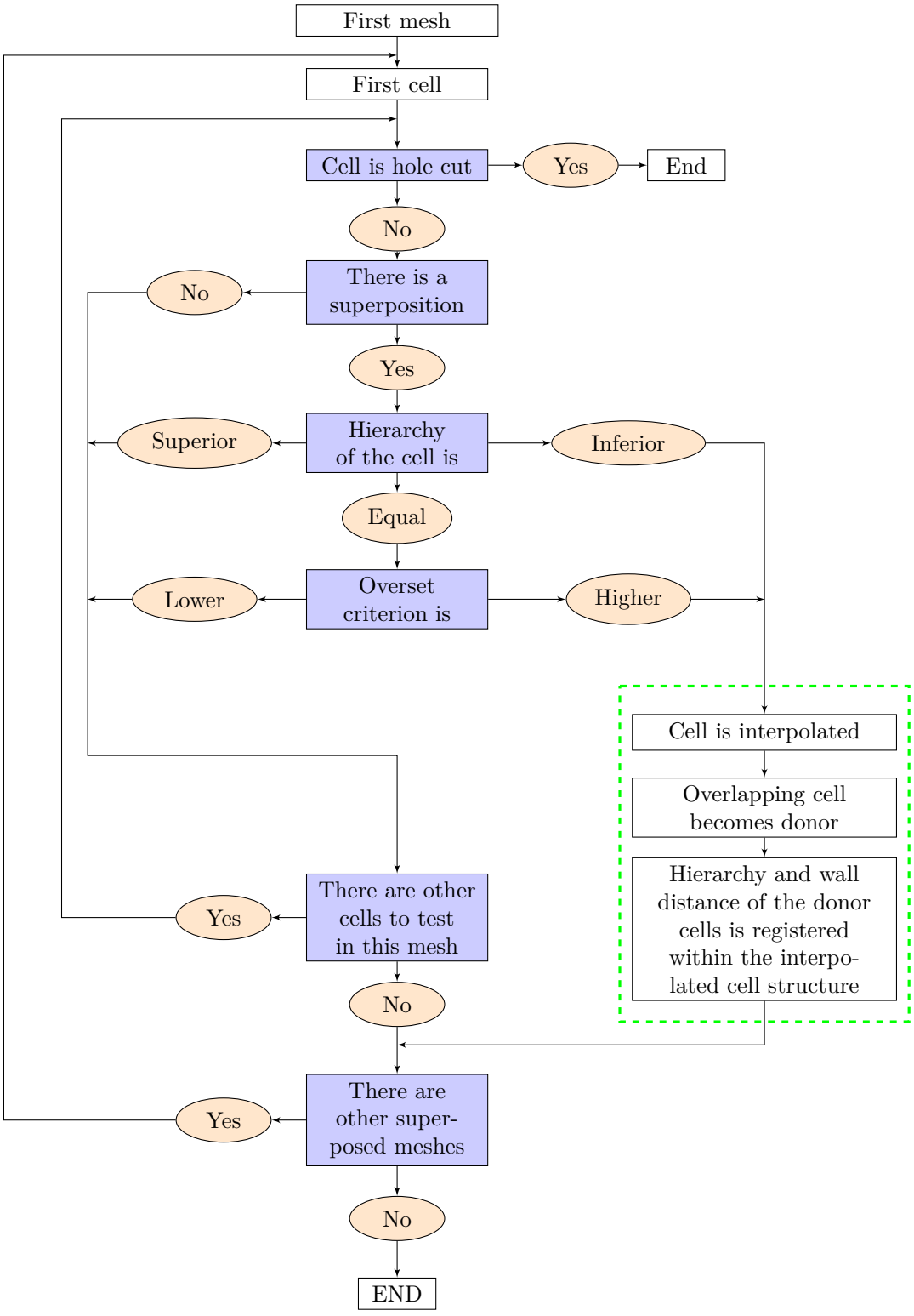
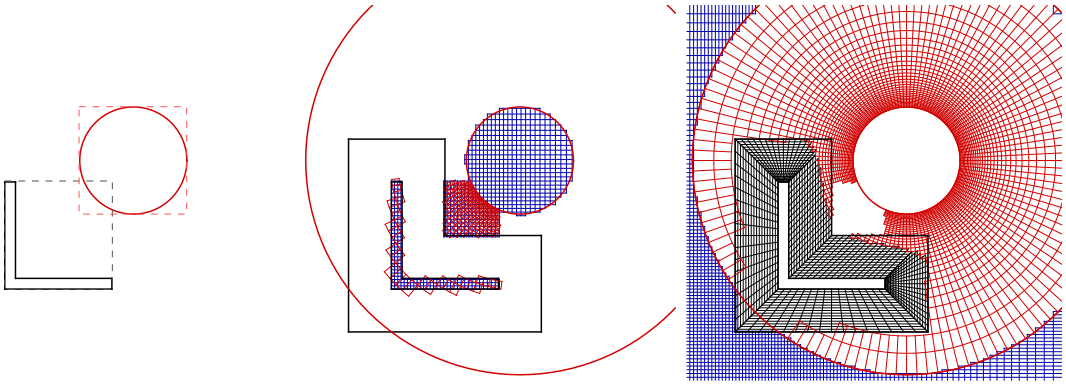


Figure 3.1 New overset identification search tree algorithm with the decoupled hole cutting

3.2 Hole cutting

With the decoupled overset identification process presented in section 3.1, a new algorithm must be implemented to proceed to the hole cutting. The algorithm already implemented in the framework could be re-used, but it is shown that it lacks robustness when facing concave geometries. This lack of robustness can also lead to a failure of the hole cutting and subsequently an improper grid assembly of the overset meshes. To highlight this, a test case was set up, consisting of a concave-shape in conjuncture with a cylinder mesh. Figure 3.2(a) shows the two geometries used, as well a their bounding boxes. In figure 3.2(b), the blanked cells by the original algorithm are shown, where it can be seen that even cells outside of the geometries are blanked. Finally, figure 3.2(c) shows the complete computed domain for this test case, and the improper hole cutting can be seen once again.



(a) Geometry (b) Hole cut cells (c) Computed domain

Figure 3.2 Concave-shape and cylinder test case studied to identify problematic hole cutting using the elimination process

Thus, a need for a better hole cutting tool is identified. In section 3.2, multiple hole cutting algorithms were detailed. Because the preprocessor is already dotted of an efficient algorithm to detect overlap between cells (Pigeon, 2015), this method can be re-used by coupling it with a CDT. The idea behind the process is that by generating a CDT of the bodies, the computational domain is completely covered: the fluid domain is covered by the provided mesh and the solid domain is covered by the CDT. The cells of the fluid domain are tested against the cells of the solid domain. If an overlap is found, the fluid cell is deemed as blanked.

3.2.1 Internal mesh

The first step to implement the new hole cutting is to compute an internal mesh of the geometry. The internal mesh is composed of a CDT based on an advancing front method. The algorithm builds on the strengths of the framework already available. The wall faces of the meshes are already ordered geometrically, since they ares used in the ice accretion process of NSCODE. Hence, the modification of this process is to simply make sure that the faces are ordered in a clockwise manner to ease the generation of the triangulation. To do so, a test on the starting face of the block is performed, to determine its orientation. Figure 3.3 illustrates how the test is performed and its signification. It can be seen that by doing the cross product of the increasing indices by the normal of the wall, the direction is automatically determined, and thus, the faces are properly ordered.

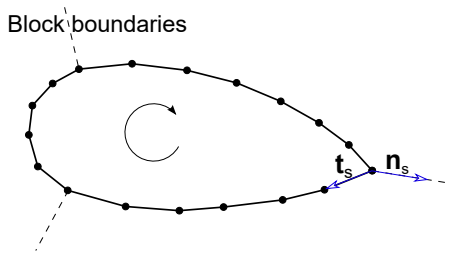
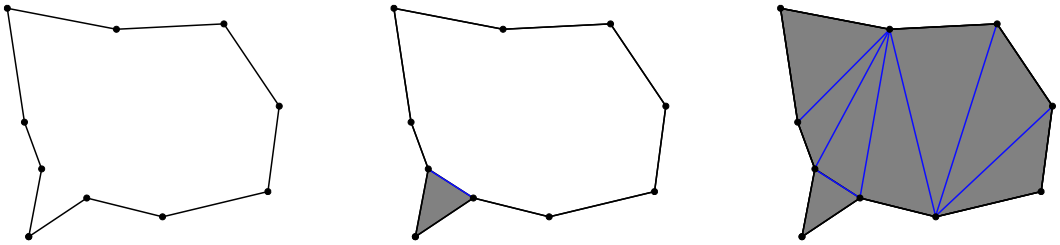


Figure 3.3 Cross product over the starting indices of the block to determine its orientation

From this list of nodes, an advancing front method based on a minimal angle criterion is used to generate the triangulation. Figure 3.4(a) shows the initial front used to compute the triangulation. The angle formed by the face is computed from it's own face and the previous face nodes. Once the face is found, a triangle is created and the 2 faces are removed from the process. In order to avoid memory increase of the initial front, one of the face is changed to the new face created, as shown in figure 3.4(b). Finally, the triangulation pursues until there are 4 faces, which are then closed by splitting them into 2 triangles. Figure 3.4(c) shows the final triangulation on the given set of faces.

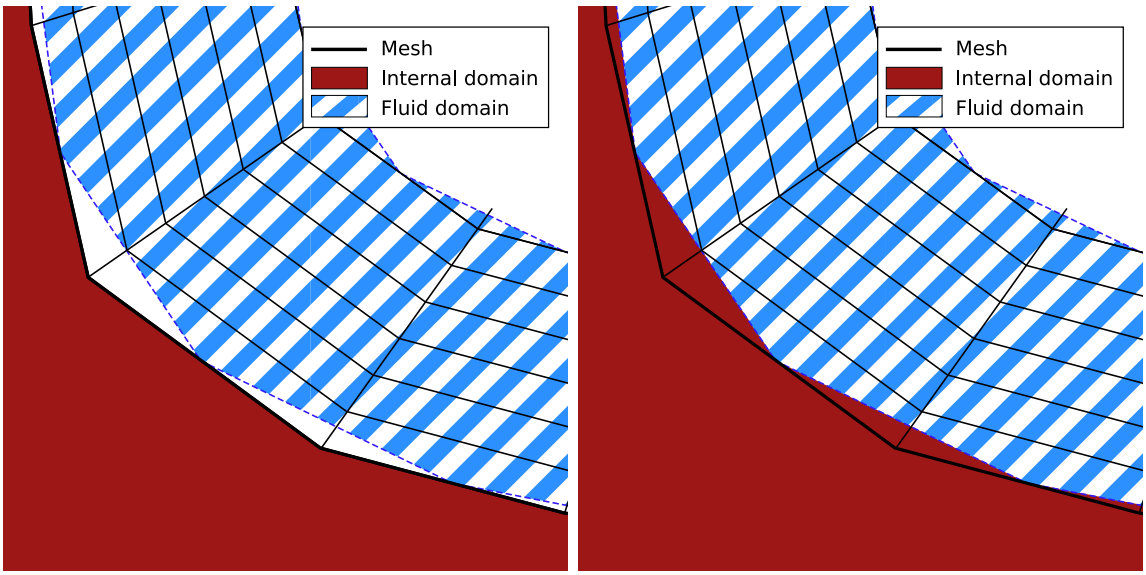
To ensure that the overset identification and hole cutting processes fully cover the whole domain, the internal mesh is performed on the face-centred coordinates. Face-centred coordinates are already computed based on the vertex-centred coordinates in the overset identification to address the corners of a block (Pigeon, 2015). Hence, it is these coordinates that are used for the internal mesh. Figure 3.5 shows the differences between the usage of vertex-centred coordinates against face-centred coordinates. It can be seen that vertex-



(a) Original front (b) Initial triangle (c) Final triangulation

Figure 3.4 Advancing front method with the original nodes connected (a), the first triangle generated (b) and the final triangulation (c)

centred coordinates produces a discontinuity between the internal and fluid meshes processes, where an overlapping cell located in the white region would fail both processes. When using the same coordinates system between the hole cutting and overset identification processes (fig. 3.5(b)), the computational domain is fully covered.



(a) Vertex-centred internal mesh (b) Face-centred internal mesh

Figure 3.5 Comparison of vertex-centred (a) and face-centred coordinates (b) used in the hole cutting, with the face-centred coordinates used in the overset identification process

3.2.2 Blanked cells detection

In the previous section, an internal mesh was generated to fully cover the *solid* domain. With this mesh, the overset identification process can be re-used, but rather than comparing fluid meshes together, the fluid meshes are compared to the internal mesh. Figure 3.6 shows the process done by the hole cutting to identify the blanked cells.

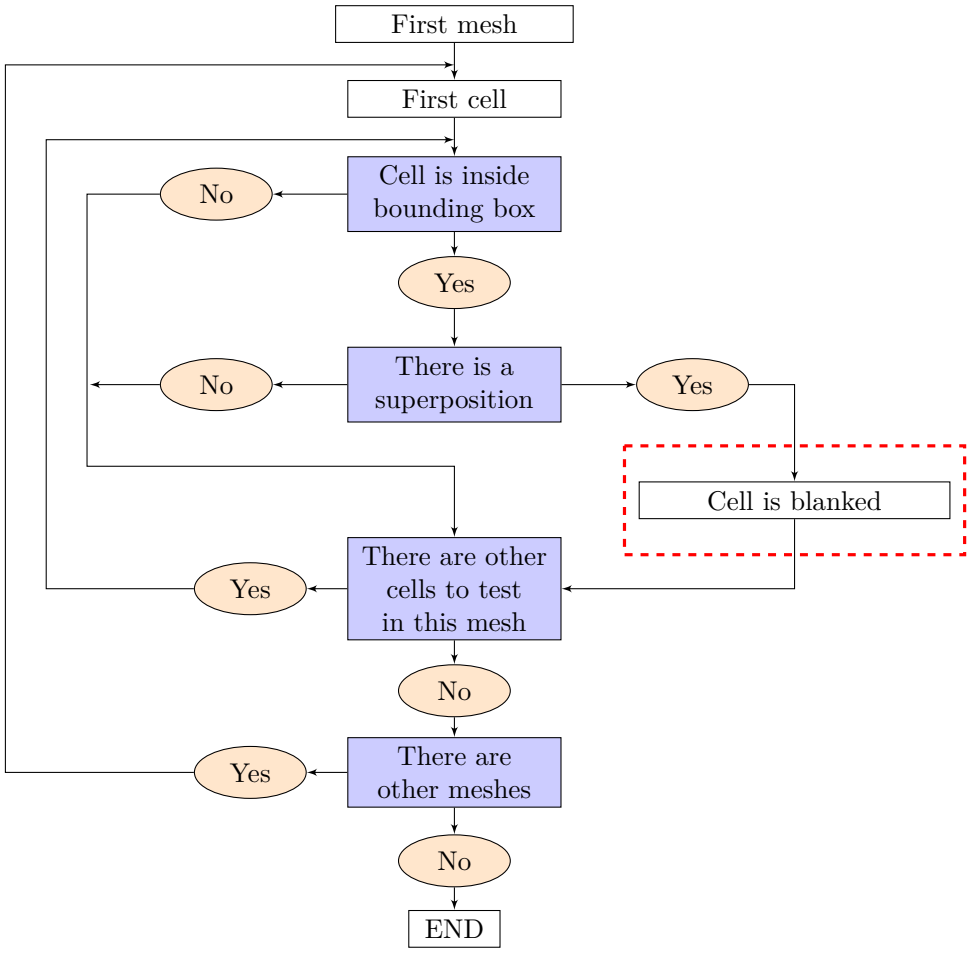


Figure 3.6 Hole cutting algorithm used to identify the blanked cells from the overset process

A first test to eliminate the cells outside of the bounding box of the geometry is used to alleviate testing cells that are by definition outside of the geometry. Then, the major part of the algorithm comes from the determination of the superposition. In this step, the cell currently tested is evaluated against the cells of the internal mesh. Once it is found to be located inside the geometry, the loop ends and the process continues to the next cell and mesh.

To determine the superposition, a test using barycentric coordinates is performed. An il-

lustration of the different variables here discussed is shown in figure 3.7. The points are all located in the same plane and every positions in this plane can be expressed from an origin using basis vectors of this plane, as shown in equation 3.1.

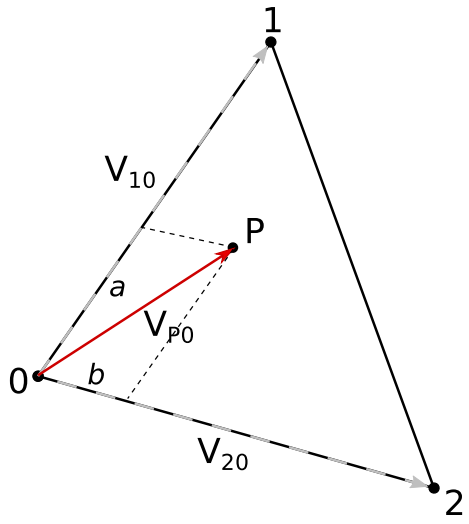


Figure 3.7 Representation of barycentric coordinates of a triangle

$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} P_{0x} \\ P_{0y} \end{pmatrix} + a * \begin{pmatrix} V_{10x} \\ V_{10y} \end{pmatrix} + b * \begin{pmatrix} V_{20x} \\ V_{20y} \end{pmatrix} \tag{3.1}$$

$$P = P_0 + a * \mathbf{V_{10}} + b * \mathbf{V_{20}} \tag{3.2}$$

$$P - P_0 = a * \mathbf{V_{10}} + b * \mathbf{V_{20}} \tag{3.3}$$

$$\mathbf{V_{P0}} = a * \mathbf{V_{10}} + b * \mathbf{V_{20}} \tag{3.4}$$

Where the indices *0, 1* and *2* refer to the three nodes of the triangle and $\mathbf{V}$ are vectors pointing between the different nodes. The origin is randomly fixed as point 0, and the basis vectors are defined from the origin to nodes *1* and 2, as shown in figure 3.7. The vector V_{P0} defines the vector between the origin and the query position – the cell-centre of the cell being tested. The coefficients a and b define how far from the origin the point is located. In this system of coordinates, the conditions to be inside the triangle are stated in equations 3.5.

$$\begin{aligned} 0 \leq a \leq 1 \\ 0 \leq b \leq 1 \\ a + b \leq 1 \end{aligned} \tag{3.5}$$

Hence, to determine if the point P lies inside the triangle, the coefficients a and b are computed from equation 3.4 and derived as:

$$a = \frac{\begin{vmatrix} V_{P0x} & V_{20x} \\ V_{P0y} & V_{20y} \end{vmatrix}}{\begin{vmatrix} V_{10x} & V_{20x} \\ V_{10y} & V_{20y} \end{vmatrix}}, \qquad b = \frac{\begin{vmatrix} V_{P0x} & V_{10x} \\ V_{P0y} & V_{10y} \end{vmatrix}}{\begin{vmatrix} V_{10x} & V_{20x} \\ V_{10y} & V_{20y} \end{vmatrix}} \tag{3.6}$$

Once the algorithm has determined that a fluid cell has an overlap with one of the triangles composing the internal mesh, its overset identity is changed to blanked. Following the hole cutting, these cells are excluded of further processes in the overset preprocessor.

3.2.3 Verification

The objective of the overset hole cutting verification takes two forms: to assess that the implemented method is able to correct improper overset grid assembly and to verify that the results computed with the new hole cutting method remain unchanged if the hole cutting was already correct. In the first case, the improper overset assembly identified in figure 3.2 is studied with the new Delaunay triangulation and the hole cutting presented. In the second case, the MDA airfoil is studied to ensure that the overset grid assembly solution is of quality in regards to the flow solution convergence and aerodynamic coefficients.

Concave geometry

In this case, the flow solution is not analyzed, it is rather the overset grid assembly that is studied. The goal is to show that the new algorithm is able to compute a proper grid assembly, even with complex geometries. As mentioned, the case previously presented (fig. 3.2(a)) is re-run with the new hole cutting method. Figure 3.8 shows the Constrained Delaunay triangulation that is used for the hole cutting process. In figure 3.9, it can be seen that the part of the cylinder mesh which was improperly identified as holecut is now correctly identified as computed. The overset grid assembly is deemed valid, because every cell has

their correct identification. This case is not subject to any flow, therefore, further validation must be performed to assess the new hole cutting algorithm.

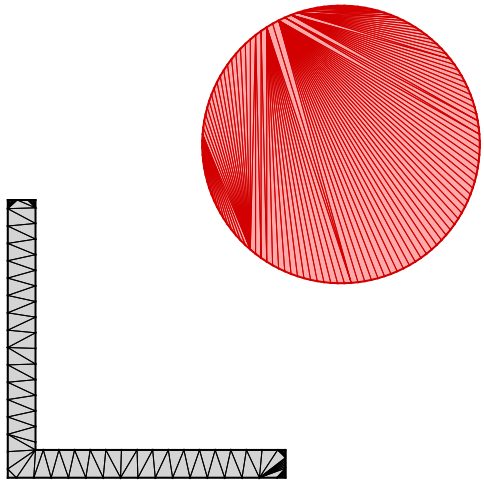
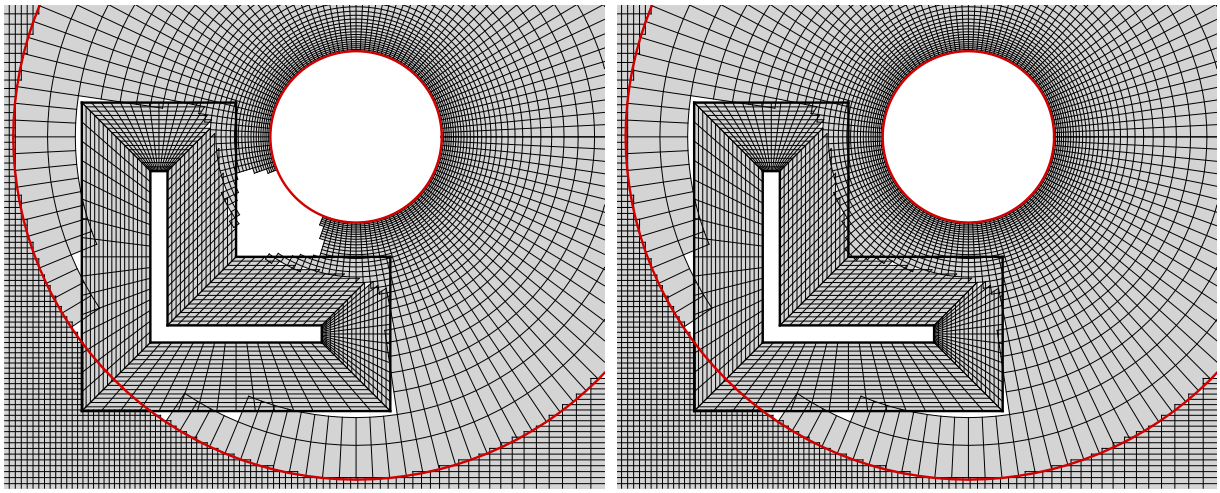


Figure 3.8 Constrained Delaunay triangulation for the concave-shape and cylinder geometry



(a) Old overset preprocessor (b) New overset preprocessor

Figure 3.9 Comparison of the overset grid assembly for the concave-shape and cylinder geometry

MDA airfoil

To complete the verification of the newly implemented hole cutting method, it is tested around a well-known airfoil, the MDA. In this case, the previous hole cutting implementation did not fail, therefore, it is important that the solution obtained with the new hole cutting method

does not affect the solution. The flow solver is set up using the parameters described in table 3.1.

Table 3.1 Parameters used for the verification of the hole cutting algorithm with the MDA case

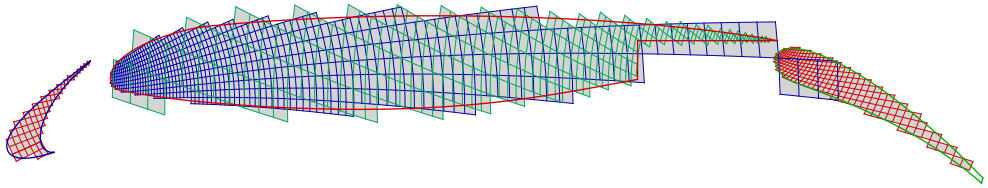
Parameter	Value
Mach Number	0.2
Reynolds Number	9.0×10^6
Angle-of-Attack	8.1°
Turbulence Model	Spalart-Allmaras
Roughness	Smooth
Solver	LU-SGS (implicit)
CFL	25.5
Dissipation scheme	Second-order matrix

First, the hole cutting from the elimination process is compared to the new hole cutting using the CDT. In figure 3.10, it can be seen that the cells identified as blanked fully cover the geometry region, which is expected for both algorithms. In figure 3.11, the overset identification of the cells is displayed, with the different elements of the airfoil (slat, main and flap) separated by colour.

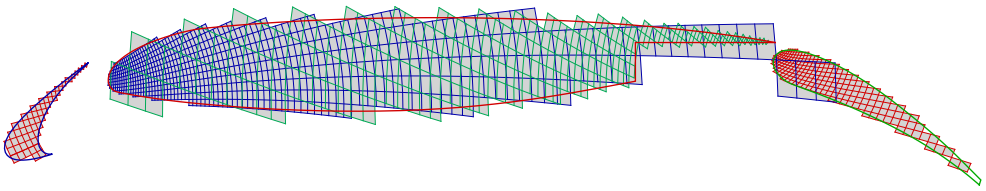
With the comparison of the overset grid assembly done, the comparison of the flow solutions can be conducted. It is to be kept in mind that the overset preprocessor has undergone an important remodeling to include the new decoupled hole cutting algorithm, as well as modifying the donor search for more accurate results. Table 3.2 shows that the aerodynamic coefficients have not been affected by the modifications. The drag coefficient is affected by an amount smaller than a drag count ($10^{-4}C_d$). For the first part of the implementation, it is important to show that they do not affect simulations that have already been verified. Thus, the developments implemented provide a more general tool that can be used with a larger variety of cases.

Table 3.2 Comparison of lift and drag coefficients between the old and new overset preprocessor

	C_L	C_D
Old overset	3.19693	0.01107
New overset	3.19277	0.01106
Difference [%]	0.13	0.08

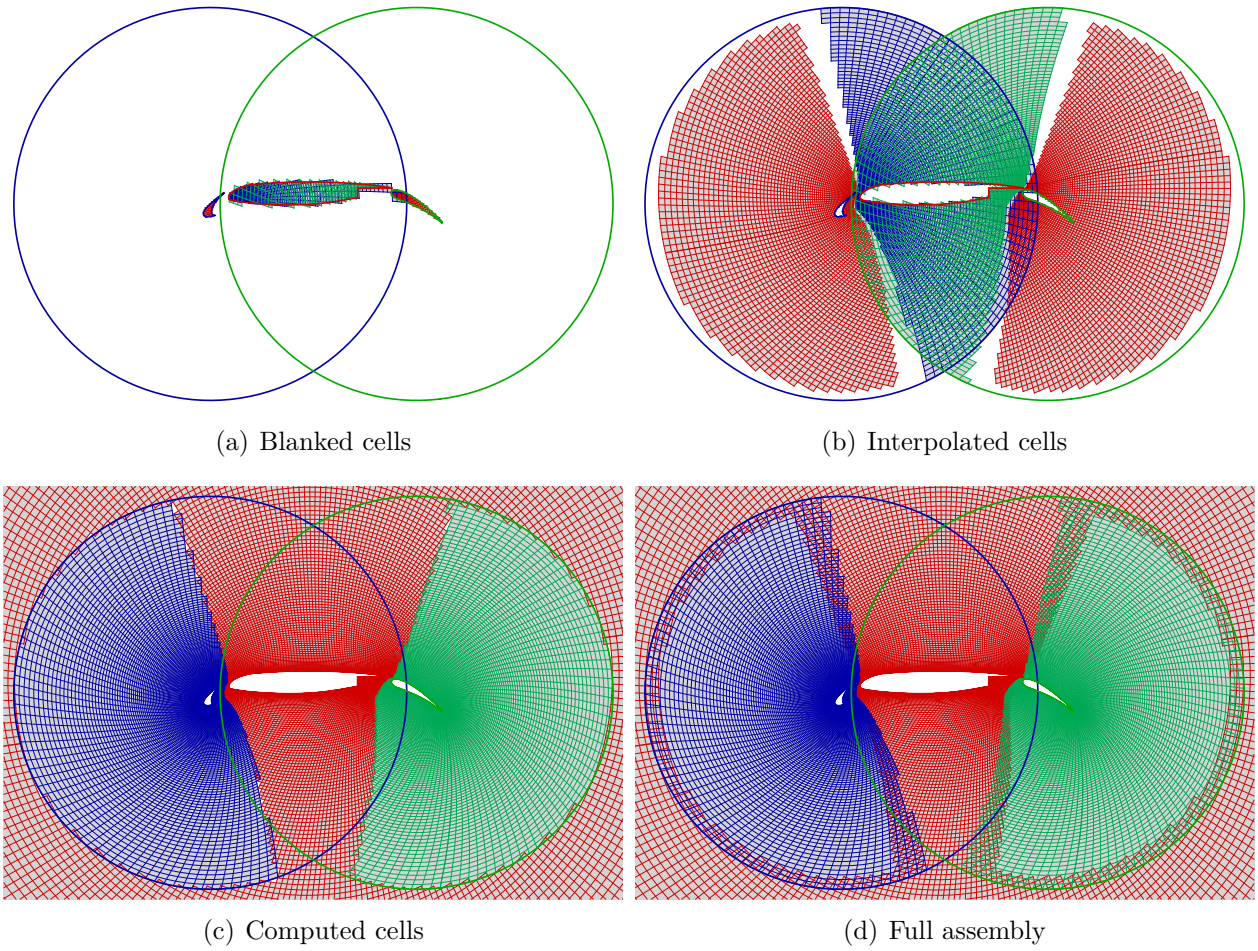


(a) Elimination process



(b) Constrained Delaunay Triangulation process

Figure 3.10 Hole cutting result for the MDA airfoil, with the old and new algorithm



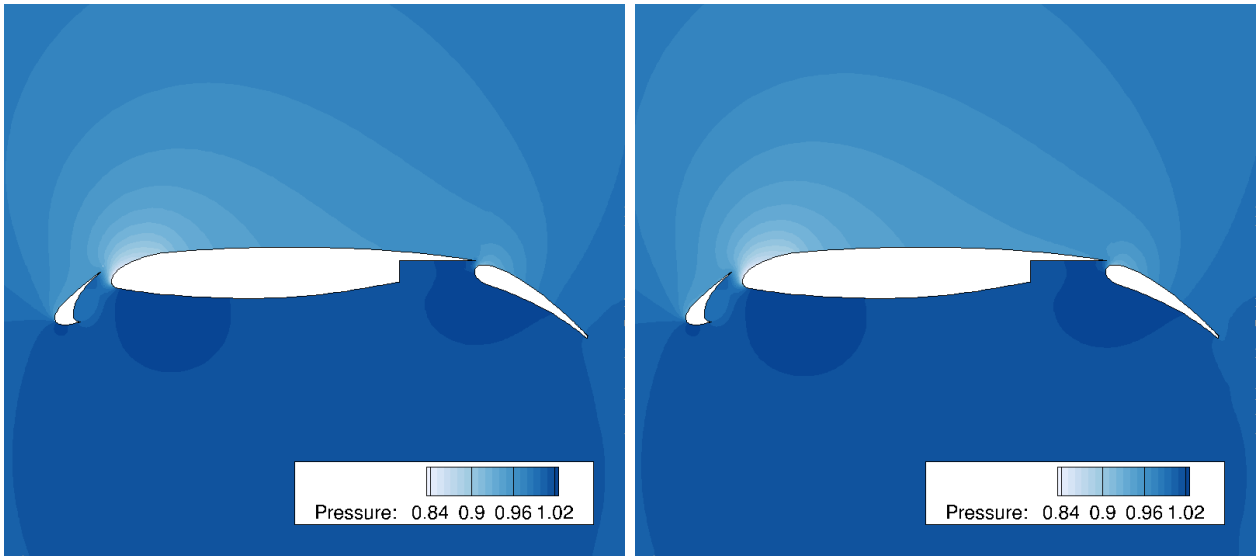
(a) Blanked cells

(b) Interpolated cells

(c) Computed cells

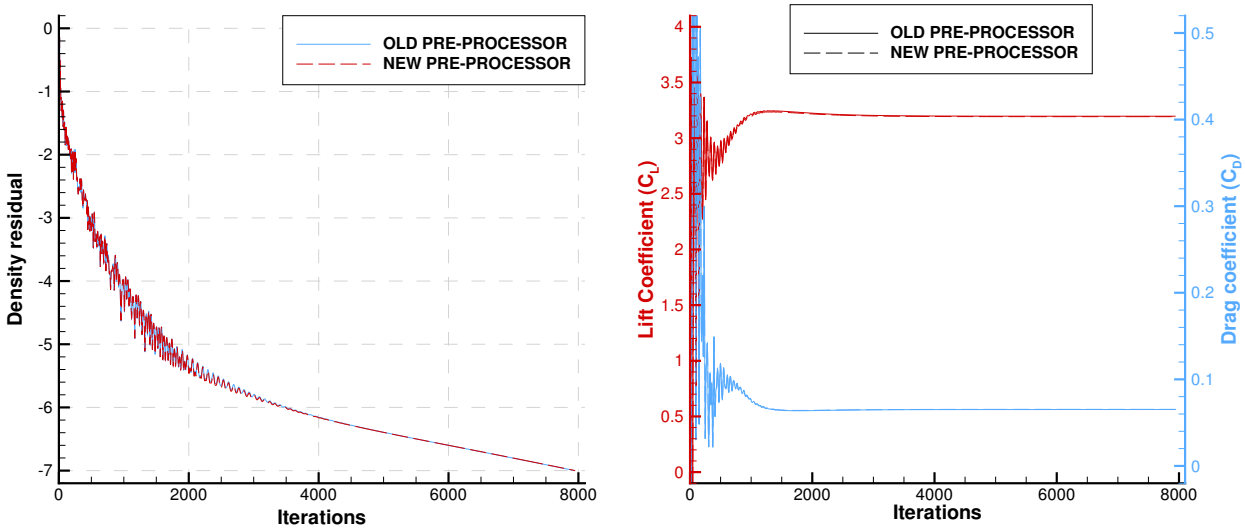
(d) Full assembly

Figure 3.11 Overset grid assembly for the MDA airfoil using the new overset preprocessor



(a) Old preprocessor (b) New preprocessor

Figure 3.12 Comparison of flow solution around the MDA airfoil for the old and new overset preprocessor



(a) (b)

Figure 3.13 Comparison of convergence around the MDA airfoil for the old and new overset preprocessor

3.3 Surface conforming meshes

A core aspect of overlapping meshes with shared body definition is the difficulty to provide an accurate interpolation in the near-wall region. This is due to surface discretization mismatch, as shown in figure 2.12, which leads to interpolation errors. To correct this, many overset processes rely on the projection of grid coordinates to ensure proper interpolation. However, the implementation of this method, while not being particularly challenging from the point of view of mathematical models, requires significant adjustments and verification that extend the scope of this thesis. Nevertheless, many other considerations are to be considered in order to treat these cases within the current framework.

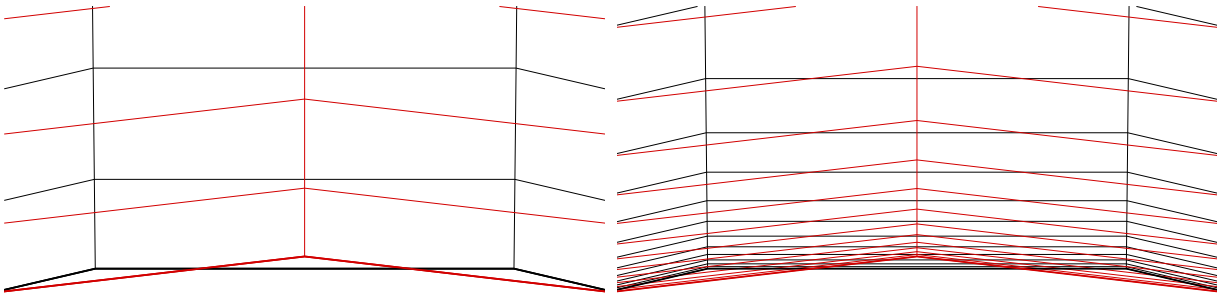
3.3.1 Effect of wall spacing

The wall spacing, which corresponds to the height of the first cell at a wall boundary, must be defined to be able to resolve the flow correctly. In the RANS formulation of the Navier-Stokes equations, it is adapted based on the Reynolds number, so that the higher the Reynolds number, the smaller the wall spacing. The surfaces studied are usually curved, but the discretization used in the FVM assumes linear discretization for each cell of these surfaces. In low Reynolds number cases, the wall spacing of the first cell is much larger than the error assumed by the linearization of the surface. Hence, the interpolation error is relatively small. However, when the Reynolds number reaches turbulent conditions and higher ($Re \leq 0.5 \times 10^6$), the wall spacing is much smaller. In these conditions, the linearization error becomes very important to consider. This difference is highlighted by figure 3.14, where it is seen that even with large discretization difference, the mesh for the low Reynolds Number case will find proper donors and the interpolation error remains small. In the mesh of the high Reynolds case, some of the cells will be improperly interpolated (some cells even lie inside the geometry and do not possess valid donors).

In these situations, special attention should be given to surface discretization. For a more general solution, a limitation is proposed in regards to collar meshes generation. When generating a collar mesh, the first and last 3 nodes (thus 2 cells) on the surface are to be extracted from the overlapping mesh. By doing so, the solution transfer between the meshes at the overset interface will have a minimal interpolation error.

3.3.2 Computational stencil correction

A particular attention was given to the computational stencil used. As previously discussed, the artificial dissipation scheme used requires to have two valid neighbors as well as the



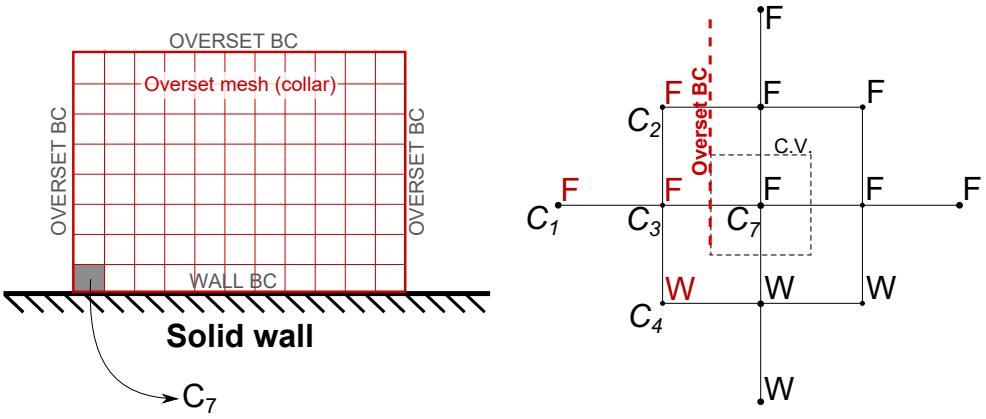
(a) Low Reynolds number (b) High Reynolds number

Figure 3.14 Comparison of low (a) and high (b) Reynolds number wall spacings with respect to relative error between the discretization of a cylinder by 1° arc lengths

diagonal neighbors. When analyzing the computed cell C_7 (figure 3.15) lying at the corner of a collar mesh (fig. 3.15(a)), bordered by an overset BC and a wall BC, this stencil can be erroneous. In figure 3.15(b), the status of the ghost cells (C_1...C_4) is uncertain. The cell C_4 can be located inside the geometry, and thus possesses no valid donors. However, as with a normal viscous wall ghost cell, this fringe cell has to be properly interpolated to make sure that the gradients – and therefore the wall properties – are correctly computed. To correct this, this fringe cell has its properties computed from the two fringe cells C_2 and C_3, as a non-slip wall BC. Hence, the wall properties are conserved in the overset BC and the fluxes computed for the corner cell will adequately reflect the physics to capture.

3.3.3 Ghost cells definition

Finally, the method of generating the ghost cells' coordinates for the overset BC is investigated. Some cases (presented in the verification section following) showed that the linear extrapolation of these cells is incorrect. As for the surface discretization, this linear extrapolation causes high interpolation error for the ghost cells of the overset BC. To satisfy a proper interpolation, the cells on the surface should be extracted from the original mesh rather than being extrapolated, similarly to the method employed for surface discretization. Figure 3.16 shows the mesh generation process used to adequately generate a collar mesh from the reference mesh. This correction ensures that the interpolation in the near wall region is adequate. When grouping these corrections altogether, a proper solution, which maintains the boundary layer of the flow, should be attained.



(a) Cell of interest (b) Initial computational stencil

Figure 3.15 Application of the computational stencil to a cell located at a corner of a mesh

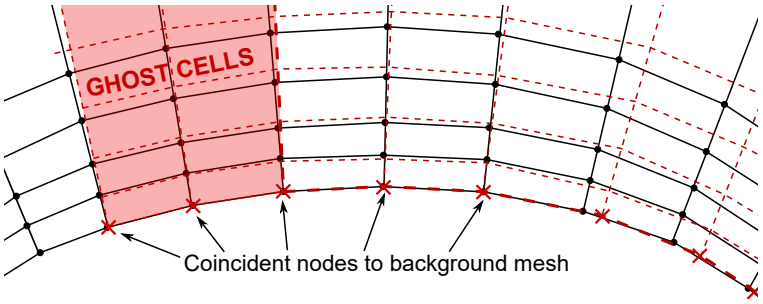


Figure 3.16 Mesh generation process for a collar mesh with coincident surface discretization about the overset boundary condition

3.3.4 Verification

The verification of the methods proposed is conducted through 2 test cases, aimed at providing a more fundamental and a more challenging cases. They consist of a laminar flow past a cylinder and a laminar flow past an airfoil. These cases aim at providing insight on the quality of the corrections proposed with respect to different flow situations.

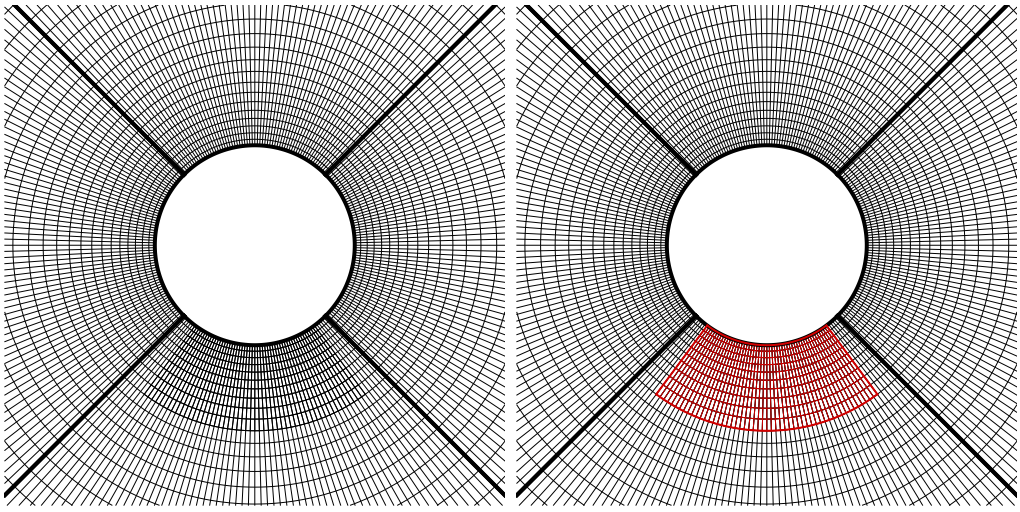
Laminar flow past a cylinder

The first case studied is a cylinder subject to a low Reynolds laminar flow. It is chosen due to its simple geometry and because wall spacing does not play an important role. In this case, the wall spacing is much larger than the discretization error. Also, the meshing technique

proposed to avoid interpolation errors near the mesh boundary can be left aside for this test case. This test case will verify the implementation of the corner boundary treatment to compute an adequate stencil. To proceed to the verification of the method, the cylinder is meshed using a multi-block mesh composed of 4 blocks, each equally spaced and expending radially. The overset configuration is composed of a small patch mesh that is added to the lower surface of the cylinder. It is extracted from one of the multi-block mesh, thus has exactly the same wall spacing and orthogonality. The solution is converged both cases with the parameters presented in table 3.3. The single and overset grids are shown in figure 3.17.

Table 3.3 Parameters used for the verification of the boundary condition treatment with the cylinder case

Parameter	Value
Mach Number	0.2
Reynolds Number	30.0
Angle-of-Attack	0.0°
Turbulence Model	none (laminar)
Roughness	Smooth
Solver	Runge-Kutta (explicit)
CFL	5.5
Dissipation scheme	Second-order matrix

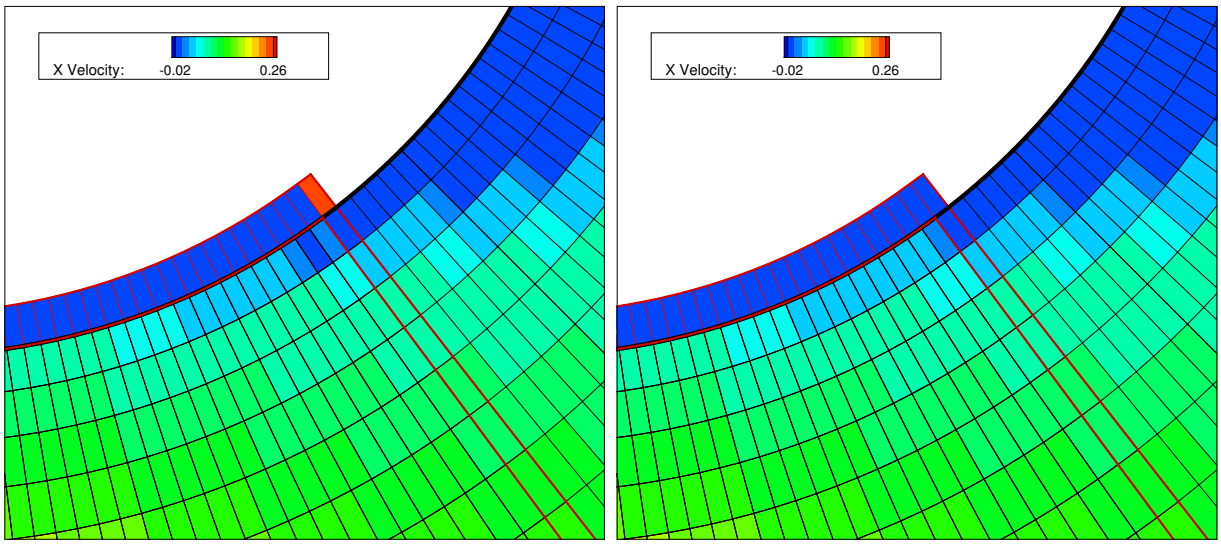


(a) Multi-block mesh (single grid) (b) Overset grid with the patch

Figure 3.17 Meshes used for the verification of the boundary condition treatment

For both the single and overset grids, the flow solutions are converged with the relative error of the density decreased by 14 orders of magnitude. A comparison of the X-velocity field

is done in the border of the patch mesh, with and without the proposed correction. Figure 3.18 shows the cell-centre X-velocity in the near-wall region, only showing the first layer of ghost cells. In the case without correction, the faulty cell is clearly visible in bright orange, which is the initialization velocity. With the correction applied, the transition appears much smoother at the mesh border. Finally, the surface coefficients are compared to have a better assessment of the correction applied. Figure 3.19 highlights the correction proposed and shows that the results obtained when applying this treatment efficiently recovers the results computed with the single grid case. This case is used as a proof of concept for the correction proposed.



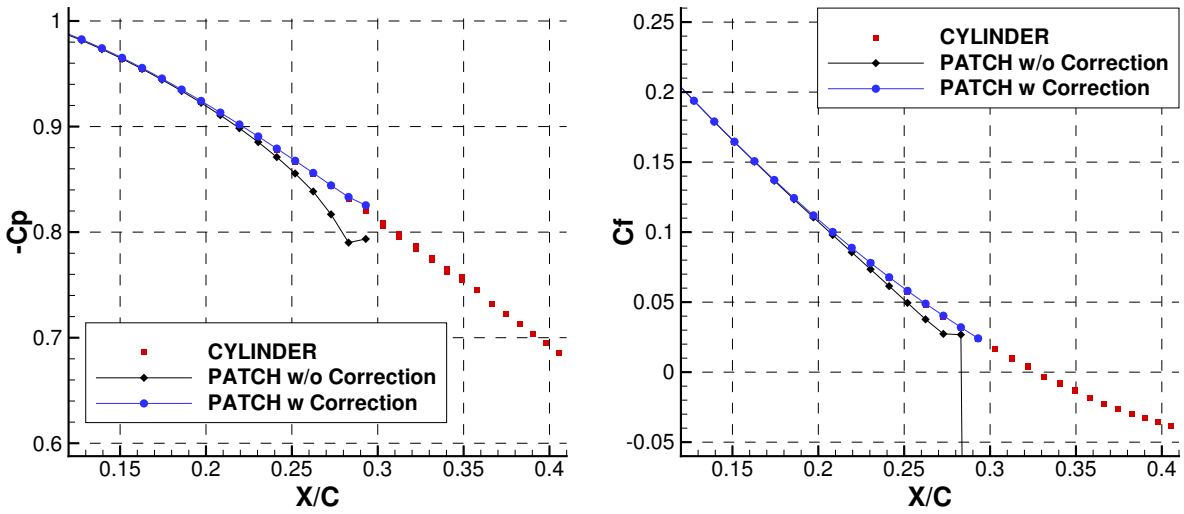
(a) Without correction (b) With correction

Figure 3.18 Cell-centre X-velocity about the patch boundary with and without computational stencil correction

Laminar flow past an airfoil

A test case which shows an increased surface discretization error is used to verify the implementation in a more realistic case. The case selected comes from a study from NASA on a NACA0012 airfoil (Swanson and Langer, 2016), whose parameters are presented in table 3.4. For this case, the new meshing technique presented previously is included to ensure that the ghost cells in the near-wall region are properly interpolated. Figure 3.20 shows the C-mesh used for the NACA airfoil (in blue), as well the the overlapping mesh (in red).

In figure 3.21, the problem caused by improper interpolation is clearly highlighted int the zoomed region. The skin friction is largely discontinued because the ghost cell inside the



(a) Pressure coefficient (Cp)

(b) Skin friction coefficient (Cf)

Figure 3.19 Pressure (a) and skin friction (b) coefficients about the the patch border

Table 3.4 Parameters used for the verification of the boundary condition treatment with the NACA0012 case

Parameter	Value
Mach Number	0.5
Reynolds Number	5000.0
Angle-of-Attack	1.0°
Turbulence Model	none (laminar)
Roughness	Smooth
Solver	Runge-Kutta (explicit)
CFL	5.5
Dissipation scheme	Second-order matrix

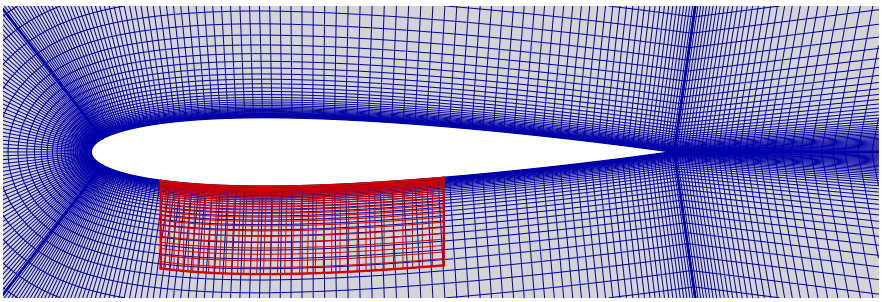


Figure 3.20 Overset mesh over the NACA0012 airfoil for the verification of the ghost cells definition

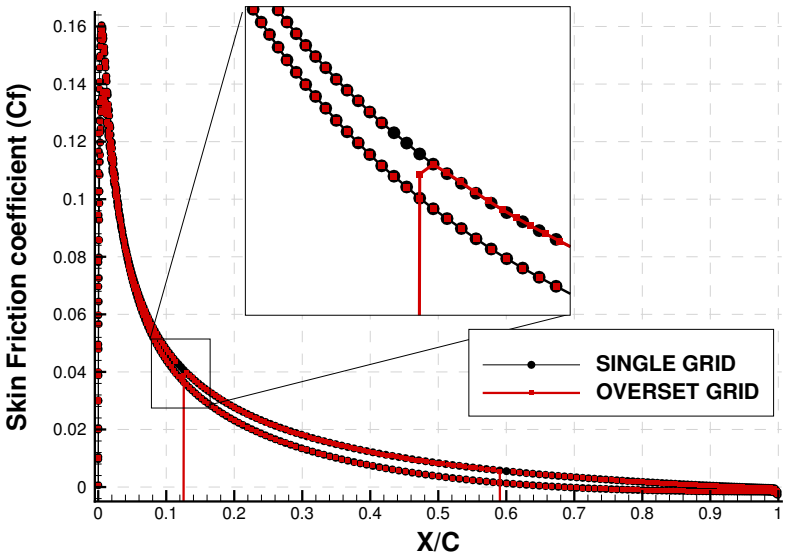


Figure 3.21 Comparison of skin friction for the single grid case and overset grid when using linear extrapolation for the ghost cells

solid wall isn't interpolated, and therefore is left to its initilization value. To verify that the methods are achieving their objective – which is to dampen the discontinuities occurring on surface coefficients – the skin friction and pressure coefficients are compared to the single grid result computed. A continuous solution is sought through the overset mesh boundary, because the presence of a boundary should not affect the results computed. The coefficients are compared against the single grid case, the regular linear extrapolation of the ghost cells, the linear extrapolation with the computational stencil correction and finally, with the user-defined ghost cells with the computational stencil correction. Figure 3.22 shows that the correction of the computational stencil, as shown in the previous verification case, leads to a closer continuous solution than without this treatment. However, there is still a kink in the solution, particularly for the skin friction coefficient. This kink vanishes when using the user-defined ghost cells rather than the normal linear extrapolation, in conjuncture with the computational stencil correction.

These two verification cases highlighted the importance of maintaining a high quality interpolation throughout the entire overset interface. It was shown that even a single cell improperly interpolated can disturb the solution and provide erroneous solutions. The correction proposed was able to ensure the continuity of the surface properties.

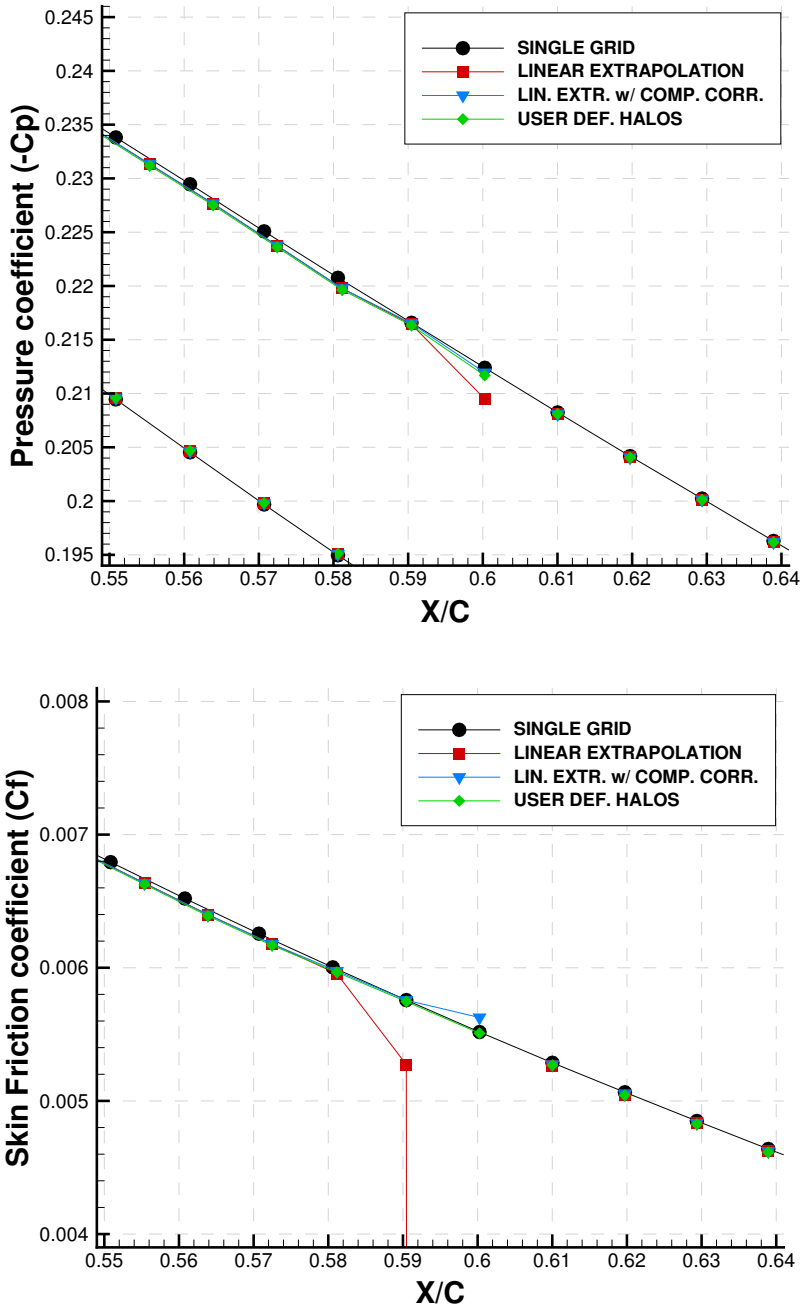


Figure 3.22 Pressure (a) and skin friction (b) coefficients for the proposed method in the overlapping region of the NACA0012 airfoil

3.4 Computation of the aerodynamic loads

Finally, an algorithm that takes into account the doubling of the surfaces must be implemented, as presented in section 2.5. In the methods presented, the Weighted Panel method

was chosen, because it alters the least with the other algorithms. The implementation of this method varies slightly from the original algorithm, by making use of the information provided by the overset process, rather than checking for overlapping faces for all of the cells.

3.4.1 Weighted Panel method

The weighted panel method is applied at the end of the overset process, after the hole cutting and overset identification is complete. Hence, the method can make use of the information provided by the previous steps of the overset preprocessor to discriminate most of the cells. Figure 3.23 shows the algorithm developed to compute the weight. The algorithm loops on all of the elements – the different geometries – and on each wall cell to determine its weight. First, the overset identification of the wall cell is checked:

- Blanked and interpolated cells must not contribute to the overall aerodynamic coefficients ($w = 0.0$);
- Computed cells without any overlap fully contribute ($w = 1.0$);
- Computed cells can be partially covered by another computed cell.

For computed cells, their hierarchy is compared to the paired cell's hierarchy. When using meshes with shared surfaces, the hierarchy of one mesh will necessarily be superior to the other, because it is required to produce a valid grid assembly (section 3.1). If the cell's hierarchy is higher, the cell is given a full contribution to the aerodynamic coefficients ($w = 1.0$). Finally, when the process has zgone through these conditions, the cells that remain are cells located near the overset interface. There, the cell can either be fully overlapped by the overlapping mesh ($w = 0.0$) or partially overlapped. In this case, the weight is determined by a ratio of distances:

$$w = \frac{\Delta s - \Delta s_{covered}}{\Delta s} \tag{3.7}$$

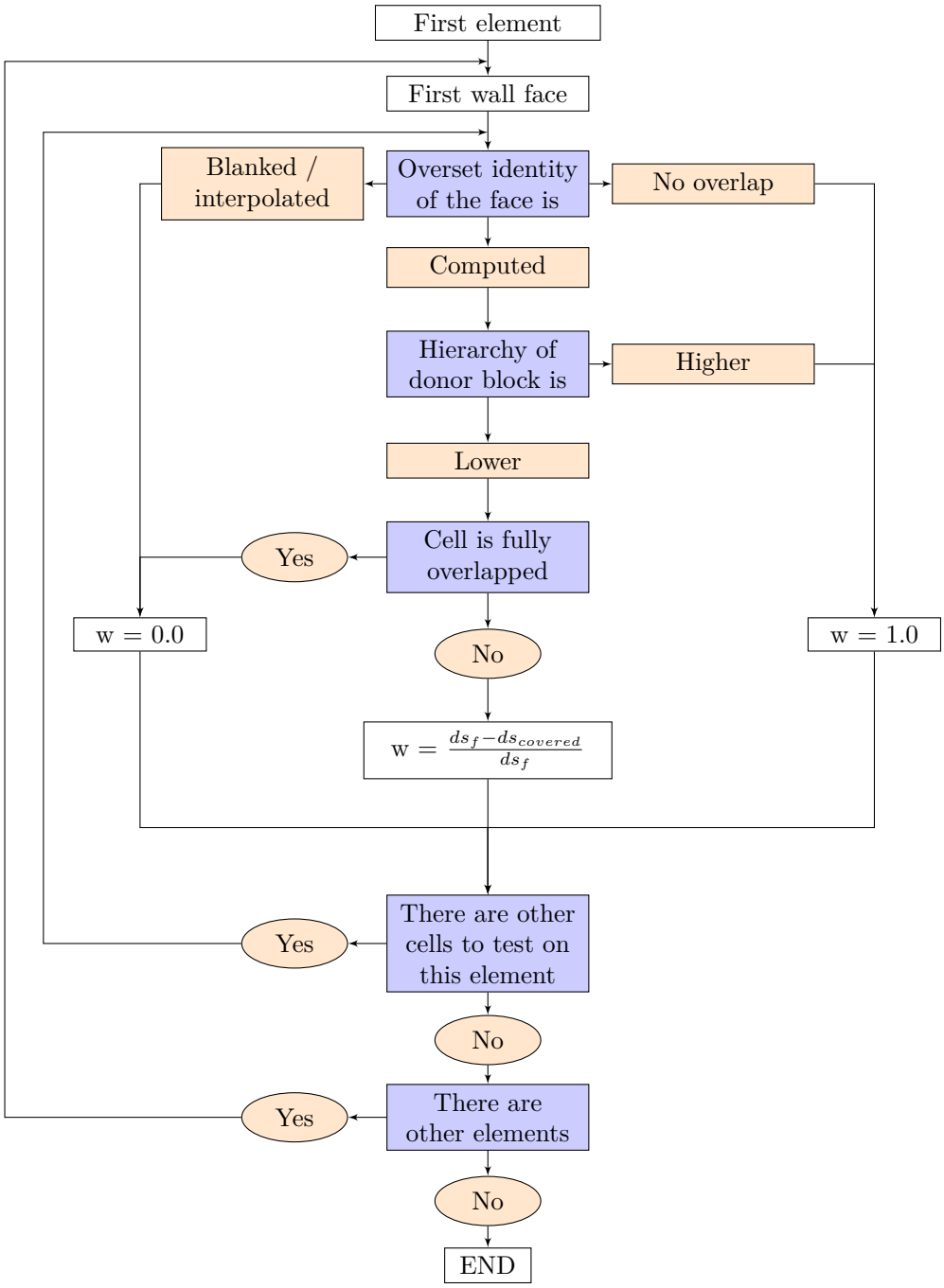


Figure 3.23 Algorithm for the computation of the weight used in the aerodynamic loads computation

3.4.2 Verification

To verify the implementation, a study on a cylinder in a laminar flow is conducted to analyze how the integration of surface properties behave with the added weight.

Cylinder

The meshes used for this verification case are the same than for the verification of the computational stencil correction (section 3.3.4). However, to avoid perfect alignment of the surface cells for both meshes, the patch mesh is rotated slightly. Because the mesh features very little interpolation error (due to large wall spacing), the overset grid assembly is still valid even if the patch's edges are not aligned on the overlapping mesh. The grid assembly is shown in figure 3.24 and figure 3.25 illustrates the result of the developed algorithm. It shows the weight as computed for the cylinder mesh, as the patch is fully computed (all the weights are equal to 1). In light grey, the cells are not overlapping any other cell, thus are fully integrated. The dark grey cells are interpolated, thus are eliminated from the surface integration ($w = 0.0$). Finally, the black cell is given a weight according to the ratio of overlap with the patch mesh – which edges are shown in red in figure 3.24.

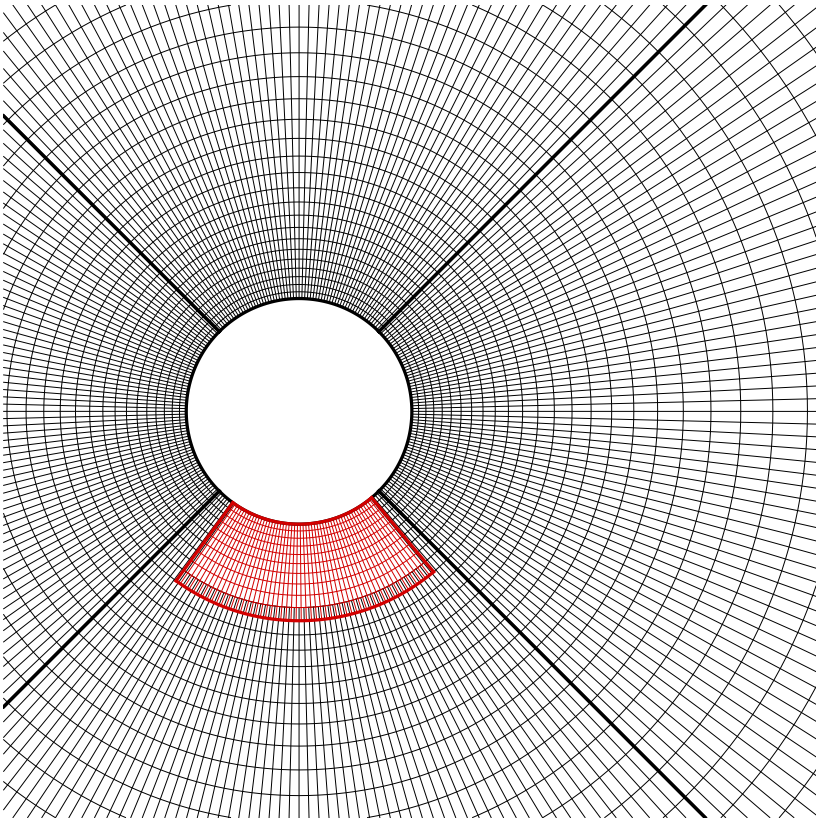


Figure 3.24 Overset grid assembly used for the verification of the weighted panel method

To evaluate the performance of this method, a grid convergence study, as proposed by Vassberg and Jameson (2010), is performed by building a family of cylinder meshes and their associated patch mesh. To maintain a similar behavior for each level of refinement, the

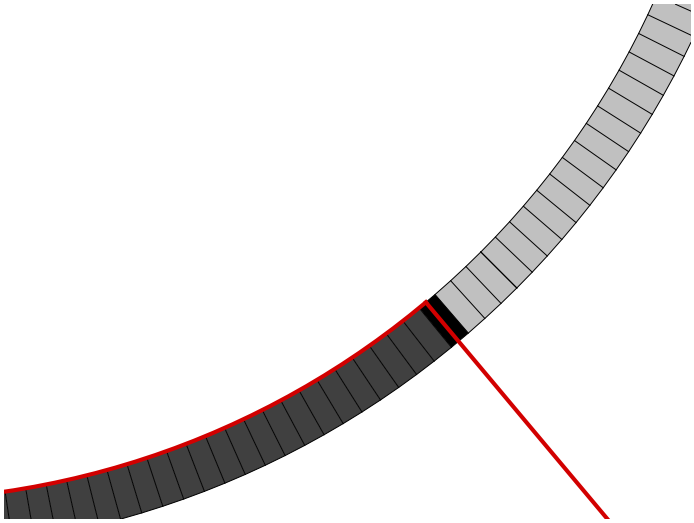


Figure 3.25 Illustration of the weighted panel algorithm result on the cylinder mesh

physical dimension of the patch is maintained constant and it possesses the same surface discretization than the cylinder mesh. Furthermore, as the grid convergence aims for results with respect to surface integration, only the surface discretization is refined. For this study, the parameter evaluated is the integrated surface length. The method used to compute the order of accuracy and the values obtained are presented in table 3.5. Although not presented here, the computation closely follows the method presented in (Vassberg and Jameson, 2010). The first column shows the number of cells discretizing the surface for the cylinder mesh. The 3rd and 5th column show the integrated surface length, $S = \sum w_i \Delta S_i$, for both the single grid and overset grid configuration. It can be seen that the continuum value obtained for both the single and overset grid is very similar, with a difference of 1.0×10^{-9}. They also converged at the same rate, showing that the weighted panel provides a mean to evaluate aerodynamic coefficients and moments to the same level of accuracy as single grids. The convergence of the integrated surface length is shown in figure 3.26, with both the order p plotted for the three finest grid refinement. With these sets of verification performed, the overset preprocessor can now be used to handle a broader range of overset meshes.

Table 3.5 Grid convergence study for the verification of the weighted panel method for the surface length integration

		Single Grid		Overset Grid	
N_{cells}	$log_{10}(1/N_{cells})$	S	$log(abs(S - S^*))$	S	$log(abs(S - S^*))$
128	-2.107	3.141276520	-3.501	3.141274900	-3.499
256	-2.408	3.141513029	-4.102	3.141512818	-4.101
512	-2.709	3.141572172	-4.700	3.141572148	-4.699
1024	-3.010	3.141586962	-5.287	3.141586960	-5.287
2048	-3.311	3.141590652	-5.831	3.141590653	-5.831
4096	-3.612	3.141591590	-6.269	3.141591592	-6.270
8192	-3.913	3.141591932	-6.707	3.141591933	-6.709
continuum		3.141592128		3.141592129	
order p		1.456		1.458	

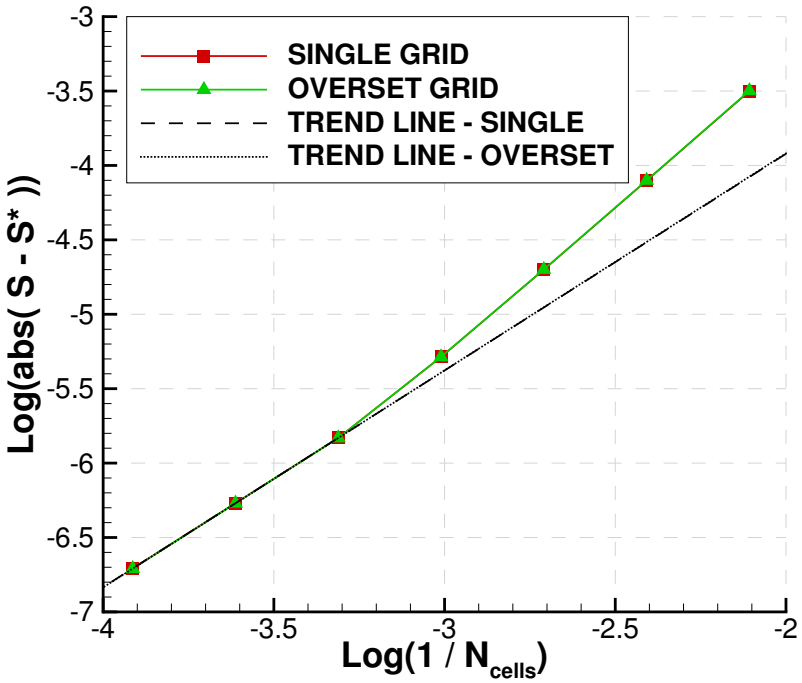


Figure 3.26 Grid convergence for the integrated surface length on a cylinder

CHAPTER 4 NUMERICAL RESULTS AND APPLICATIONS

The objective of this chapter to show the new capabilities of the overset grid preprocessor to handle complex geometries. The applications are selected to show the value of the overset capabilities with respect to industrial-type cases such as transonic cases and aileron deflections.

4.1 Mesh generation

As the overset preprocessor is by definition a mesh treatment tool, a special concern must be given to mesh generation. Throughout this thesis, the meshes presented have all been generated with the idea that they would be used in conjuncture with other meshes. This mind-set aims at providing meshes that will communicate efficiently, such as avoiding large discrepancies in cell sizes at the overset interface and ensuring a sufficient overlap in geometry proximity regions.

Two mesh generation software were used: NSGRID and ANSYS ICEM-CFD. NSGRID is an in-house mesh generation tool, developed within the research group of Professor Eric Laurendeau (Hasanzadeh *et al.*, 2016). This software features a multi-block structured mesh generation using a transfinite algebraic algorithm and an hybrid elliptic-parabolic smoother. ANSYS ICEM-CFD is a commercial mesh generation tool that offers great flexibility, and for the concern of this thesis, the capacity to generate multi-block structured meshes. In this thesis, most of the meshes are generated using NSGRID, as it is easier with this software to control the surface discretization when using surface conforming meshes. Also, the meshes present a better orthogonality at wall boundaries, due to the smoothing applied. In some of the cases – where the case presented an increased topology difficulty – ICEM-CFD was used.

4.2 Laminar flow

4.2.1 Definition of the case

The first case to be presented in this chapter is one already shown: the cylinder in a laminar flow. Whereas it was previously used to verify the implementation of the boundary treatment (see section 3.3.2), this case is now used to assess the robustness of the flow solver NSCODE. Specifically, this case seeks to highlight that the flow solver NSCODE is not affected by the new implementations.

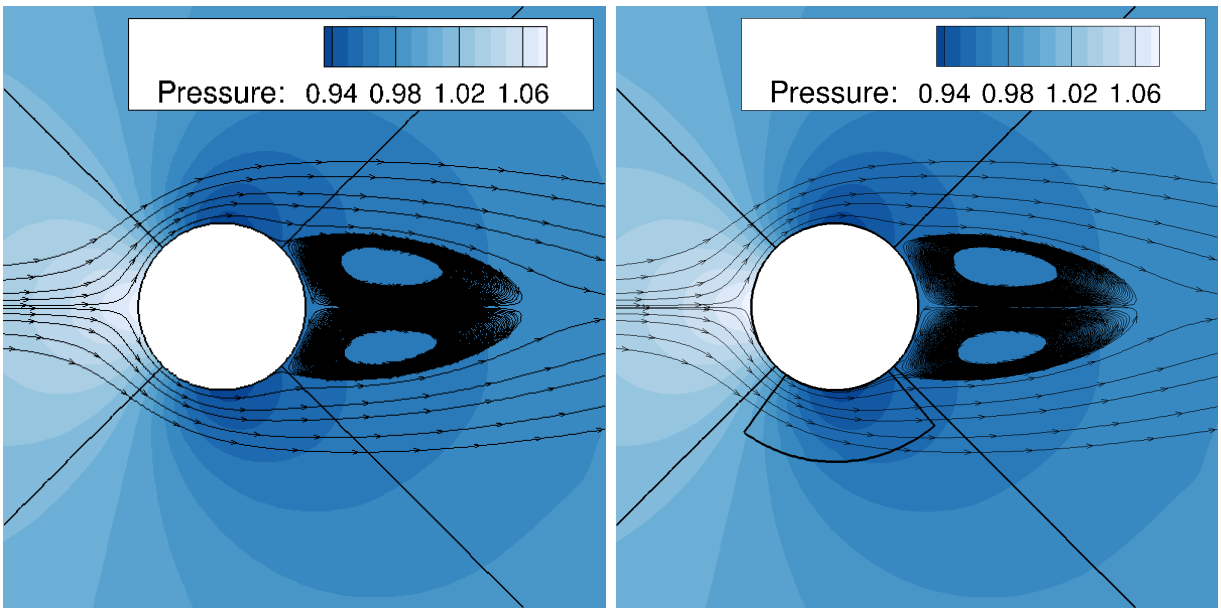
The parameters used for this case are the same than previously presented in table 3.3, for the verification of the boundary treatment. The cylinder is subject to an incompressible laminar flow ($Ma = 0.2$) at a Reynolds number of 30. In this configuration, the flow solution is expected to be in a steady state, with two recirculation zones behind the cylinder. The mesh used is also the same than the one presented in the verification of the computation of aerodynamic loads (section 3.4). The mesh for the single grid simulation is comprised of 4 blocks of 64 by 64 cells. The far-field boundary is located at 50 cords from the center of the cylinder. For the patch mesh, it is a single block with 54 by 10 cells, with the same surface discretization than the single grid mesh.

4.2.2 Analysis

In figure 4.1, it can be seen that the flow solution preserves its symmetric property, even with the non-symmetric patch mesh. Figure 4.2 shows that the convergence of the single and overset grids are similar, when compared without the multigrid acceleration technique. The overset grid converges in approximately 250 iterations faster than the single grid. It is to note that in the current framework, the multigrid technique cannot be used with overset meshes. These results show that the new developments made in the flow solver to treat this new category of meshes did not disrupt the performance of the flow solver, with regards to convergence. Also, when comparing the pressure and skin friction coefficients (figures 4.3(a) and 4.3(b)), it can be seen that the solutions are on top of one another and that the solutions are continuous through the patch boundary. Finally, the drag coefficient for both grid configurations is compared (table 4.1), which shows a difference of 0.21% for the pressure drag (C_{dp}) and 0.01% for the friction drag (C_{df}). These results show that the accuracy and the robustness of NSCODE is maintained when using surface conforming meshes.

Table 4.1 Drag comparison for the cylinder in laminar flow

	Single	Overset	Difference [%]
C_d	1.8329	1.8304	0.13
C_{dp}	1.1770	1.1745	0.21
C_{df}	0.6560	0.6559	0.01



(a) Single grid (b) Overset grid

Figure 4.1 Streamlines and pressure contours past a cylinder for a single (a) and overset grid (b)

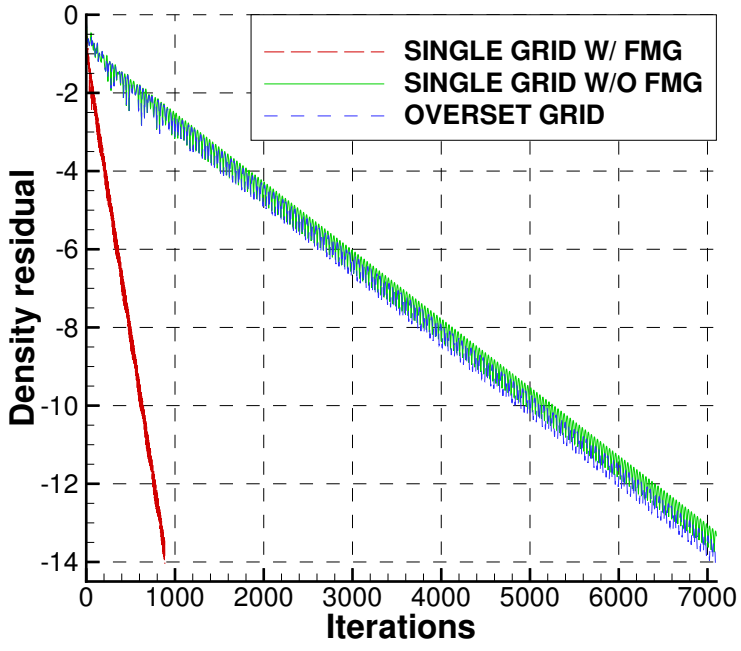
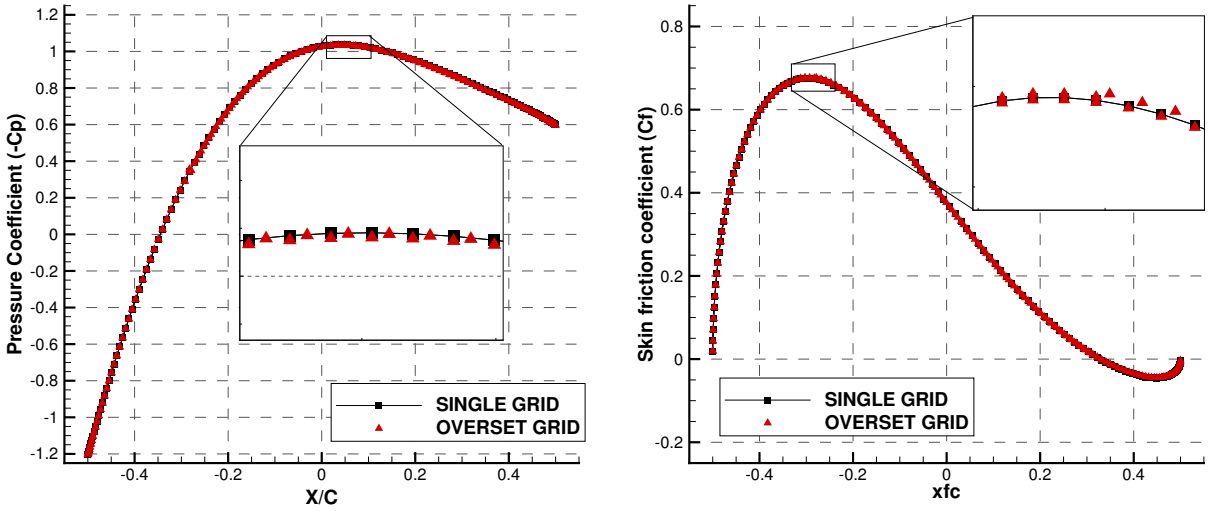


Figure 4.2 Convergence for the cylinder subject to a laminar flow for the single and overset grids



(a) Pressure coefficient

(b) Skin friction coefficient

Figure 4.3 Pressure (a) and skin friction (b) coefficients comparison for the cylinder subject to laminar flow for the single and overset grids

4.3 Transonic flow

4.3.1 Definition of the case

The second application is a RAE2822 airfoil study. This case is subject to a transonic flow with comparison from the experimental results of the case 9 of Cook *et al.* (1979). This study aims at showing the efficiency of the overset method in an industrial context, where simulations in the transonic regime are commonly performed for civil and business aircrafts. The simulation is run at $Ma = 0.73$ and $Re = 6.5 \times 10^6$. The experimental angle-of-attack (AoA) is 3.19° and the numerical case is run at a constant value of C_l of 0.8033, which is the experimental lift coefficient. The numerical AoA found is 2.7225° for the overset grid case. For this simulation, the Spalart-Allmaras turbulence model with uniform roughness is used. A summary of the case's parameters can be found in table 4.2.

The airfoil mesh is a 4 blocks O-mesh, each with 64 by 128 cells, with the far-field boundary stretching to 50 cords. The collar mesh is a single block of 96 by 64 cells, with an increased point distribution density at the shock location ($x_{shock} \approx 0.55c$). The overset grid assembly is shown in figure 4.4. The wall spacing around the airfoil is defined to ensure that the dimensionless wall distance is below 1 (for this mesh, $y^+_{max} \leq 0.45$).

Table 4.2 Parameters used for the transonic flow past the RAE2822 airfoil

Parameter	Value
Mach Number	0.73
Reynolds Number	6.5×10^6
Angle-of-Attack	2.7225°
Turbulence Model	Spalart-Allmaras
Roughness	Smooth
Solver	Runge-Kutta (explicit)
CFL	5.5
Dissipation scheme	Second-order matrix

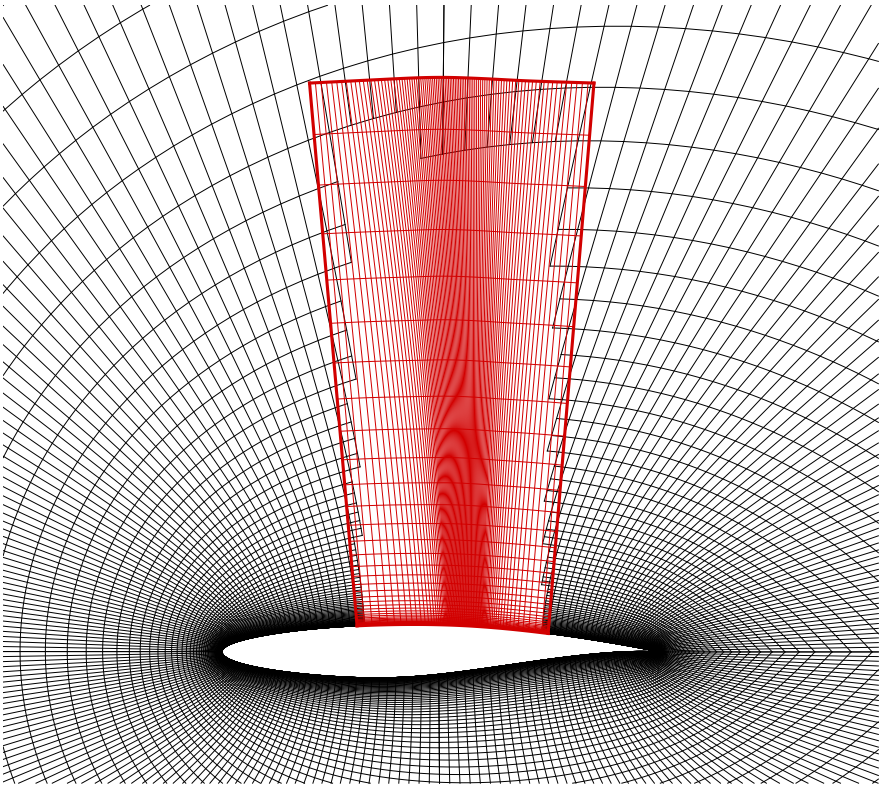


Figure 4.4 Overset grid assembly of the RAE2822 airfoil

4.3.2 Analysis

Here, comparison of the convergence (fig. 4.5) for both configurations (single and overset grids) show similar behavior, without the usage of the multigrid acceleration method. Figure 4.6 shows the pressure contours of both the single (fig. 4.6(a)) and overset (fig. 4.6(b)) grid solutions, showing a noticeable increase of the shock resolution for the overset grid solution. Table 4.3 highlights the new calculation of the aerodynamic coefficients. From these results,

it can be seen that the forces are properly computed. The most probant results for this simulation arise at the comparison of the surface properties (figure 4.7). When comparing the pressure coefficient distribution, the increased accuracy of the overset solution is clearly shown in the supersonic shock region, which also lies much closer to the experimental data. For the skin friction coefficient, the increased accuracy of the shock resolution is also shown by a much steeper cutoff.

One issue that arises with this test case that did not appear in the previous cases is the presence of discontinuities in the skin friction coefficient. The two discontinuities occur at the transition between the collar mesh and the airfoil mesh. As it can be seen, this disruption of the skin friction does not occur on the pressure distribution. One thing to consider between these two parameters is that the pressure coefficient is a dimensionless variable computed from the pressure variables and the far-field properties:

$$C_P = \frac{P - P_\infty}{\frac{1}{2}\rho_\infty V_\infty} \tag{4.1}$$

whereas the computation of the skin friction coefficient relies on both the primitive flow variables (the viscosity μ for instance) and the velocity spatial derivatives. In the context of this thesis, the interpolation in wall vicinities has focused on the interpolation of the primitives, but these results show the need to emphasis on the evaluation of the derivatives near boundary conditions.

Table 4.3 Comparison of the aerodynamic coefficients for the RAE2822 airfoil

	Single	Overset	Exp.
α [°]	2.7225	2.7225	3.19
C_l	0.793533	0.803356	0.8033
Δ C_l [%]	1.216	0.007	——
C_d	0.016704	0.016926	0.0168
Δ C_d [%]	0.571	0.750	——
C_m	-0.091905	-0.093676	-0.100
Δ C_m [%]	8.095	6.324	——

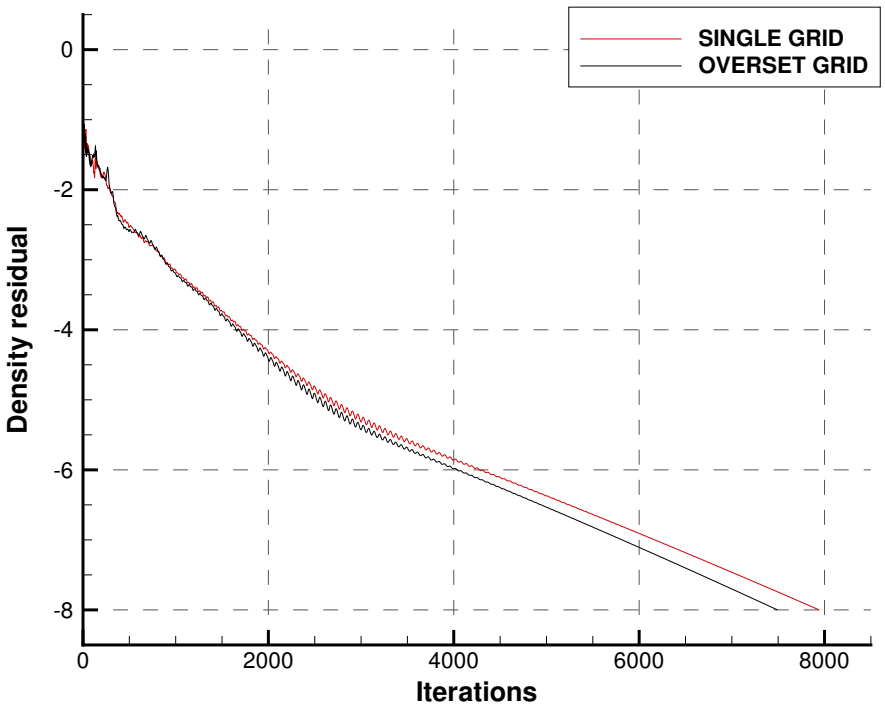
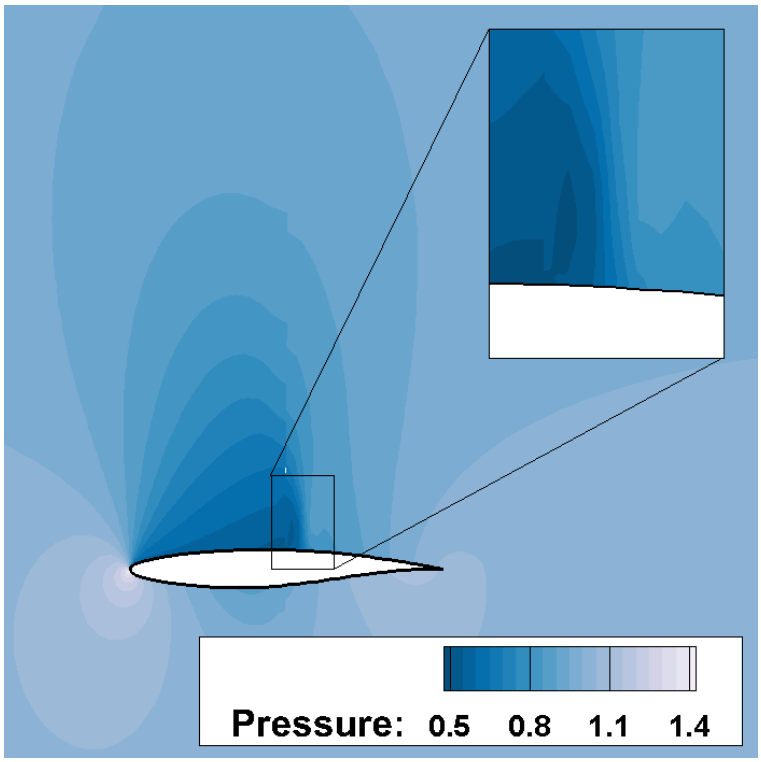
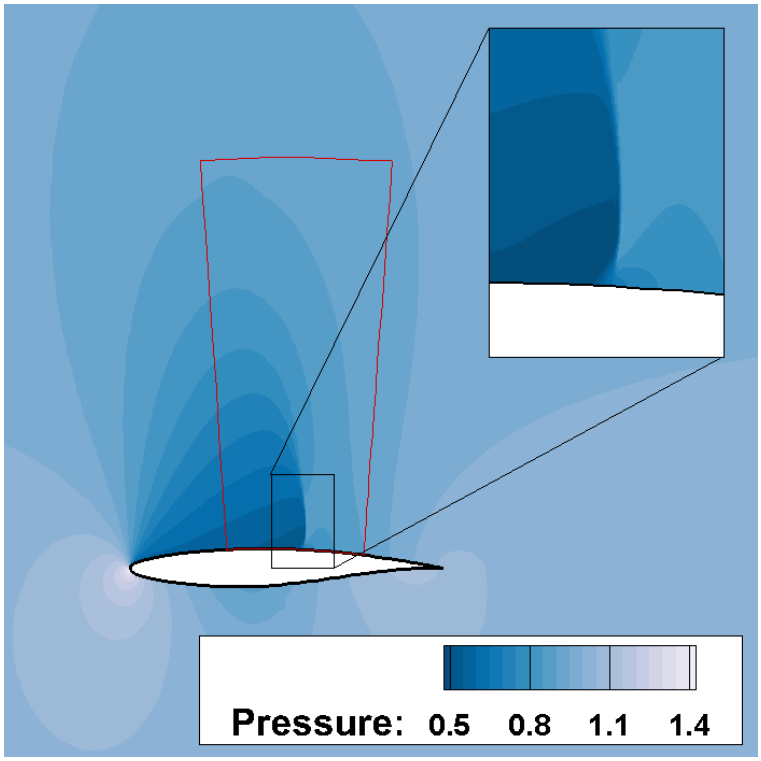


Figure 4.5 Convergence of the density residual for the single and overset grid configurations

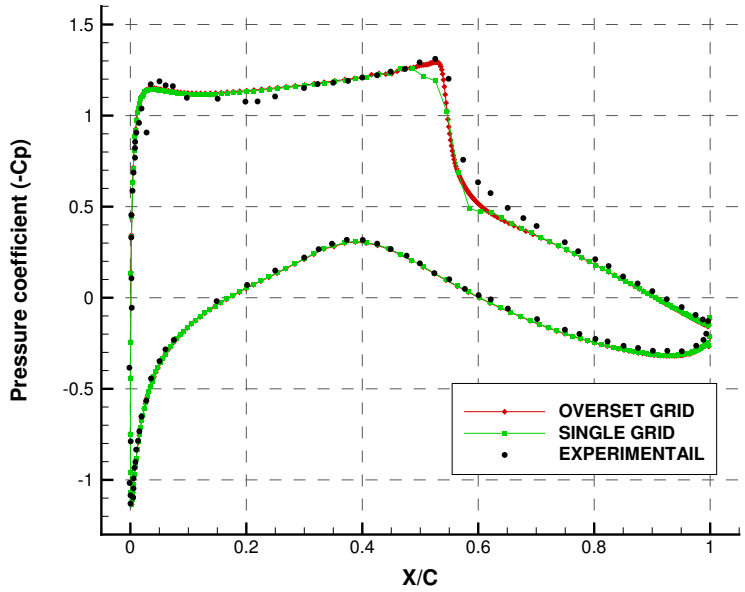


(a)

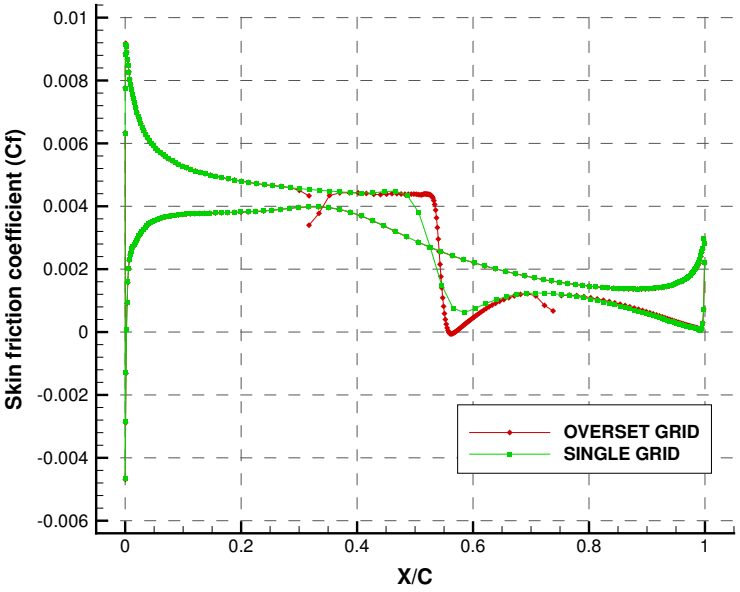


(b)

Figure 4.6 Pressure contours of the RAE2822 airfoil for the single (a) and overset (b) grid solutions, at $Ma = 0.73$, $Re = 6.5 \times 10^6$ and $\alpha = 2.7225^{\circ}$



(a) Pressure coefficient



(b) Skin friction coefficient

Figure 4.7 Pressure (a) and skin friction (b) coefficients of the RAE2822 airfoil for the single and overset grid solutions, compared to experimental data

4.4 Steady wing-aileron

4.4.1 Definition of the case

The final test case aims at testing the limits of the overset preprocessor with respect to interpolation of flow properties near wall boundary layer. In this application, a wing-aileron configuration is studied to evaluate the performance of the overset preprocessor in configurations with close proximity between geometries. Typically, wing-aileron configurations do not have an opened gap between the aileron and the main element – the gap would be closed by seals. However, one of the strengths of the overset method is to simulate applications with geometries in relative motions to one another. Thus, the gapped geometry is more suitable for unsteady CFD simulations, to compute the lift derivate ($C_{L\dot{\alpha}}$) value of the aileron for instance. The test case selected is a NACA0012 airfoil with a thick trailing edge. The parameters for this case are selected based on the validation case presented on NASA's Turbulence Modeling Resource website (Rumsey, 2016). It serves as validation for the turbulence model, but it is here used in the context of a gapped wing-aileron configuration. The parameters used are presented in table 4.4. The aileron is obtained by cutting the airfoil with a circular gap centred at $3/4c$, with a radius equal to the height of the airfoil at that location. This gap configuration is designed to maintain the geometry as close as possible to the original airfoil. The NACA 4-digit airfoil family have an analytical definition of their coordinates, given by (for symmetrical airfoils in the series NACA00XX):

$$y = \pm\frac{t}{0.2}[0.2969(x)^{1/2} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4] \tag{4.2}$$

where t is the relative thickness of the airfoil, which is 0.12 for a NACA0012 airfoil. At the aileron centre, this yields a radius of 0.0316. For the wing element, the same technique is employed, by defining the cutting radius to: $r = r_{aileron} + g$, where g is the gap size. Three different gaps have been derived: a large gap ($g = 0.01c$), a medium gap ($g = 0.005c$) and a small gap ($g = 0.0004c$). The geometries are presented in figure 4.8. In this study, the three gapped geometries are compared to a clean configuration (without gap) using a one-to-one multi-block mesh and the new developments presented throughout this thesis. These configurations are evaluated at three aileron angle: 0°, 5° and 10°.

The far-field boundary is located at 50 chords from the origin and the wall spacing is selected to ensure that the dimensionless wall distance is below 1. For the single grid, the large and medium gapped main element and the aileron, the meshes are O-mesh type topologies. For the small gap, a different mesh topology was used for the main element. On the two other

Table 4.4 Parameters used for the NACA0012 airfoil in the wing-aileron configuration

Parameter	Value
Mach Number	0.15
Reynolds Number	6.0×10^6
Angle-of-Attack	0.0°
Turbulence Model	Spalart-Allmaras
Roughness	Smooth
Solver	LU-SGS
CFL	25.0
Dissipation scheme	Matrix Dissipation

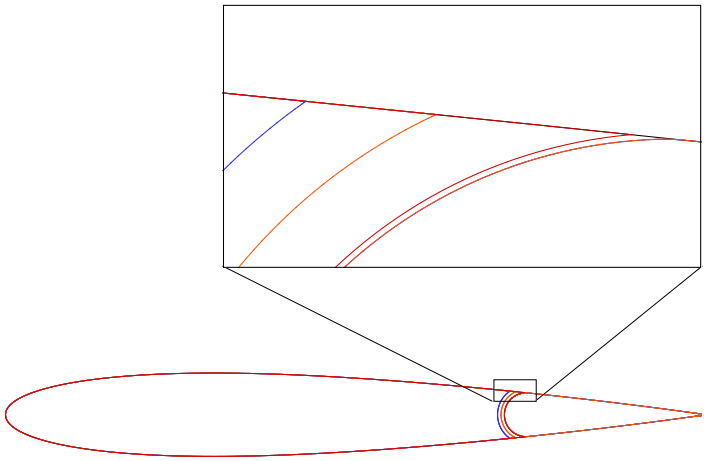


Figure 4.8 NACA0012 airfoil with the wing-aileron configuration

wings, the trailing edge maintained a relatively low sharpness, which made it possible to generate an O-mesh topology around the wing. However, for the small gap, the wing trailing edge is extremely sharp, which is poorly suited for such a topology. To provide an higher quality mesh, a dual C-type mesh topology was used, as shown in figure 4.9.

This mesh also highlights one of the strengths of the overset method, which is to hide low quality cells in the blanked region of the mesh. Because blanked cells are not computed, little concern may be given to mesh quality in such regions. In this particular case, the priority is given to provide high quality cells in the gap region and around the edges of the wing. However, this configuration makes it hard to maintain an appropriate grid quality throughout the entire meshed domain. Figure 4.10 shows that in order to maintain a high quality in the computed domain, the stretching pulls the cells into the area occupied by the aileron. Orthogonality and stretching is maintained to appropriate levels near the main

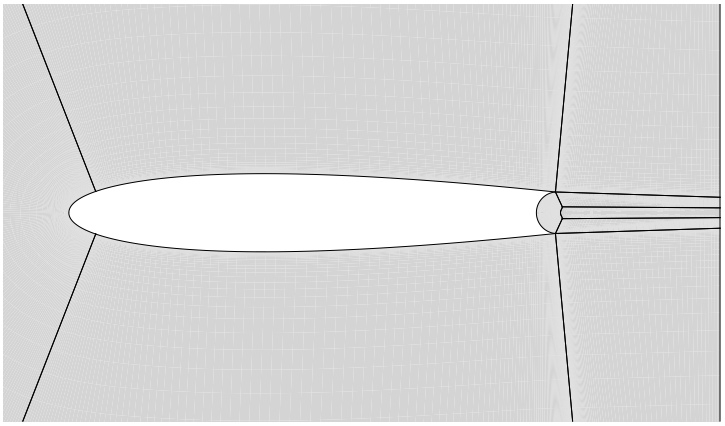
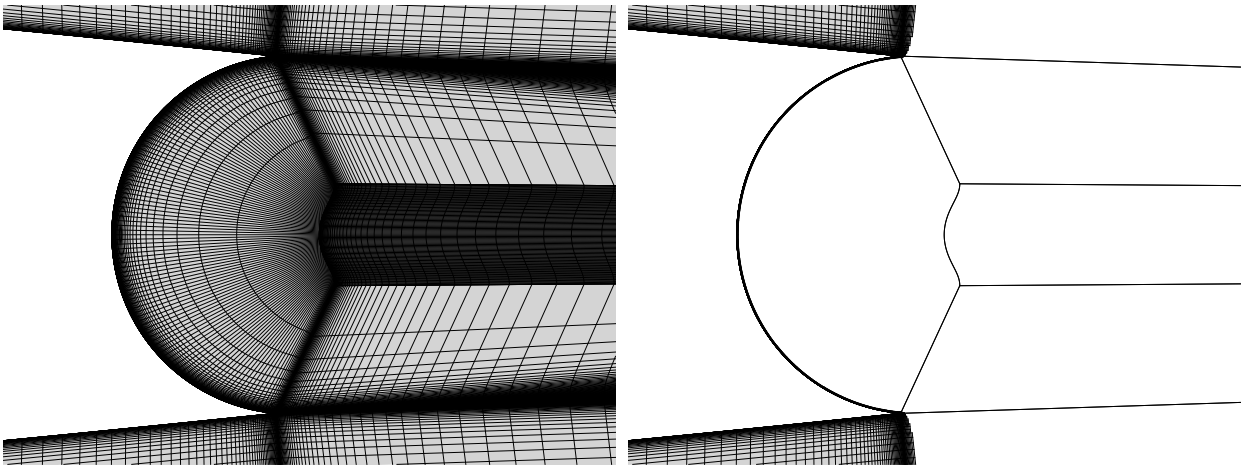


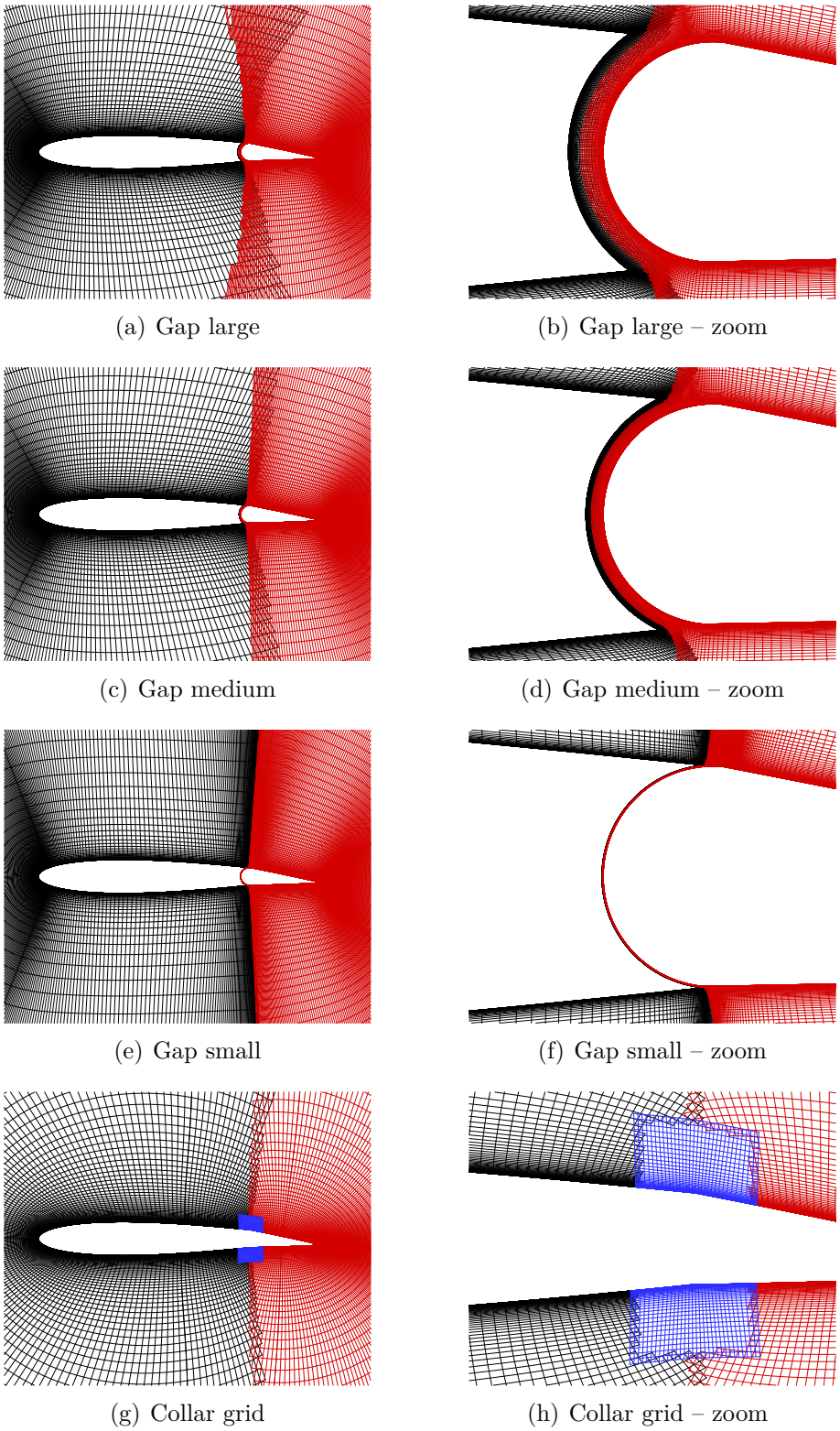
Figure 4.9 Dual C-type topology used for the small gap wing mesh

element and this leads to high aspect ratio and skewed cells near the back of the airfoil, as shown in figure A.2. In this region, the stretching ratio increases up to 2, while good meshing practice keeps the stretching below 1.2. Finally, the presence of the wing sharp edges forces a mesh surface refinement of the aileron. Near these edges, the mesh of the aileron needs to be denser to ensure a valid grid assembly in that region. The grid assemblies for the three gapped configurations as well as the grid assembly of the collar case, for the aileron deflection angle of 5° are shown in figure 4.11. The grid assemblies of the other configurations are shown in appendix A, which also includes more figures from the current study. The appendix A also includes the single grid meshes used for the three deflection angles (figure A.1).



(a) Poor quality cells (b) Blanking applied

Figure 4.10 Dual C-type mesh generated around the main element with the poor quality cells in the aileron vicinity (a) and with the blanking applied (b)



(a) Gap large

(b) Gap large – zoom

(c) Gap medium

(d) Gap medium – zoom

(e) Gap small

(f) Gap small – zoom

(g) Collar grid

(h) Collar grid – zoom

Figure 4.11 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 5.0°

4.4.2 Analysis

To evaluate the efficiency of the method, the four overset configurations are compared to the single grid results. All of the simulations were converged with the error on the density residual reduced by 8 orders. Also, lift and drag coefficients convergences were obtained on all test cases.

The surface properties, namely the pressure and skin friction coefficients, are compared to the experimental data from Gregory and O'Reilly (1970) in figure 4.12. For the 0° deflection, the results compare very well with the experimental data. However, the gap size has a large influence on the surface coefficients. At the gap location, the large gap produces a large difference in the coefficients. Also, it is to note that the same kinks in the C_f distribution than previously observed in the transonic case appear when looking at the collar grid results.

The aerodynamic forces (C_L and C_D) and moment (C_M) are presented in table 4.6. Figure 4.13 puts each of the coefficients in comparison with the single grid results. The small gap results are close to the single and collar grid results, with a difference to the single grid results of respectively 1.3% and 3.1% on the lift and drag coefficients. As the aileron deflection angle increases, it can be seen that the large and medium gap configurations get farther and farther away from the baseline results, with a difference at the higher deflection angle of respectively 31.0% and 17.7% on the lift coefficient.

To explain this behaviour, a closer look must be taken into the flow solutions computed. For the large and medium gaps at the high aileron deflection (figure 4.14), a large mass flow passes from the lower to the upper surface through the gap, which is expected. This acceleration of the flow around the aileron leading edge causes fluctuations in the pressure and surface coefficients (as presented in figures A.3 and A.4. With the smaller gap, this behaviour fades away and the difference with the clean configuration result is 0.2% on the C_L . An evaluation of the mass flow rate ($\dot{Q}$) is performed at the mid-point of the gap ($y = 0$), which is defined as:

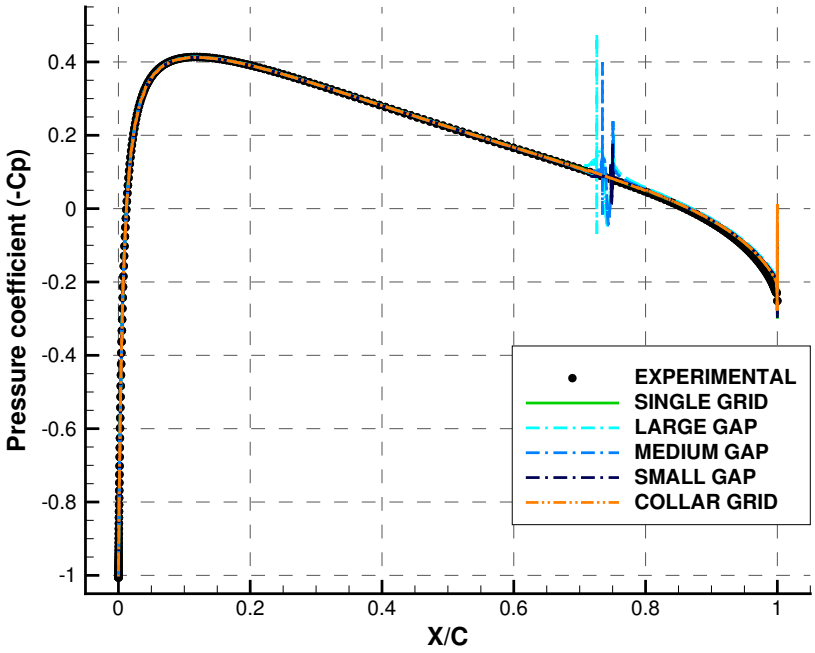
$$\dot{Q} = \rho V A = \oint_A \rho V \, dA \tag{4.3}$$

Results are presented in table 4.5. Between the large and medium gaps, the mass flow rate decreases with a rate slower than the decrease in area (1.61 against 2 for the area). However, when comparing the small and medium gap, the flow rate decreases faster than the difference in area (19.2 against 12.5). With the small gap, the boundary layers of both geometries are close to one another, leading to a highly viscous region. This has the effect to clog the gap, leaving a small amount of flow to cross from the lower to the upper surface.

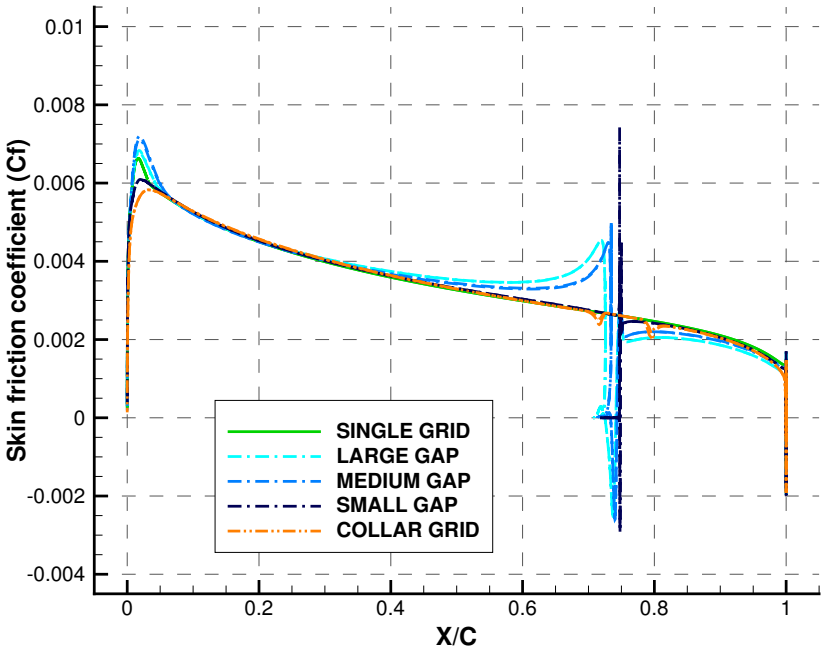
Table 4.5 Mass flow rate in the gap region for the aileron deflection of 5.0°

Gap	Size [m]	$\dot{Q}$ [kg/s]
Large	0.01	1.968E-02
Medium	0.005	1.220E-02
Small	0.0004	6.346E-04

Finally, the collar grid method was tested to showcase mesh generation in the case of bodies in relative motion. For the three deflection angles, the same meshes for the wing and aileron were used, and only the collar meshes needed to be regenerated with the appropriate aileron deflection. This proved to be a good solution, as these meshes are easier to generate than a full multi-block mesh. However, for each deflection angle, the surface points were extracted manually from the wing and aileron surfaces. From these points, the new collar meshes were generated. This procedure is not suited for moving meshes or for a large number of deflection angles. More work would be required to provide automation for such simulations. Literature shows that tools are developed to provide automation of collar meshes generation (Péron *et al.*, 2014). Another solution has also been developed through the software Engineering Sketch Pad (Haimes and Dannenhoffer, 2013), which has been coupled to an automatic mesh generator.

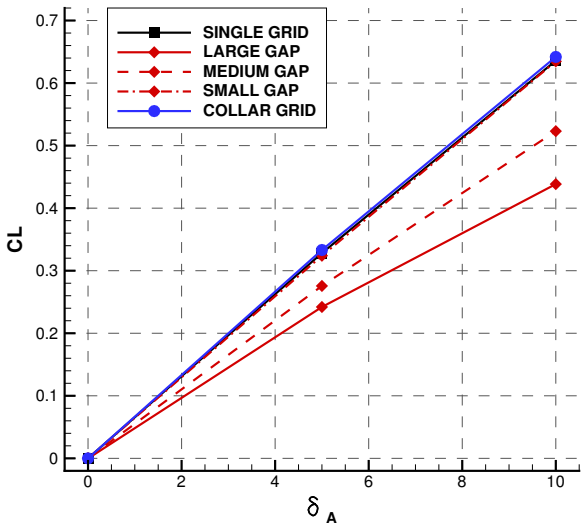


(a) Pressure coefficient

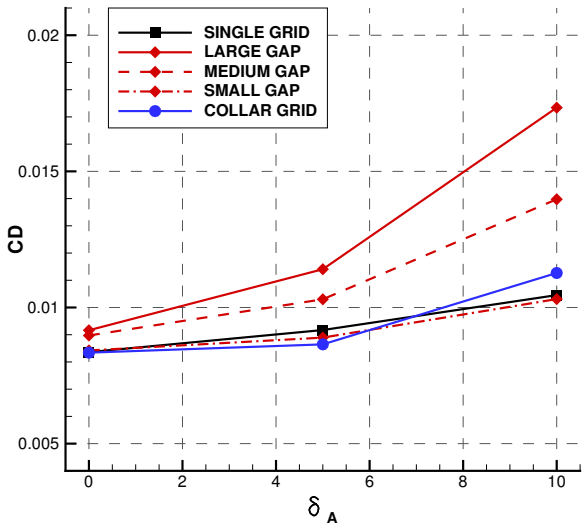


(b) Skin friction coefficient

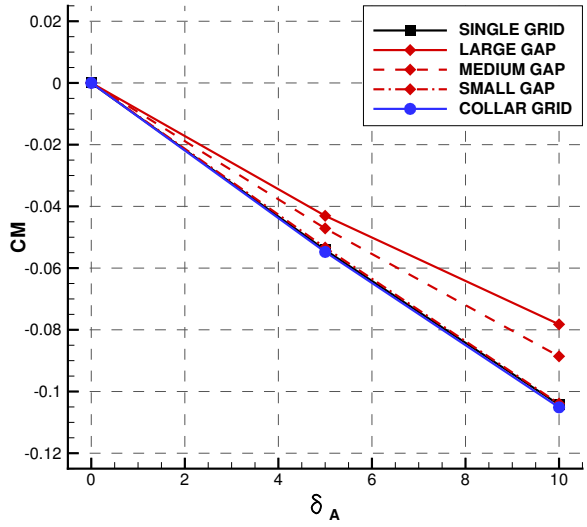
Figure 4.12 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 0.0°



(a) CL versus δ_A

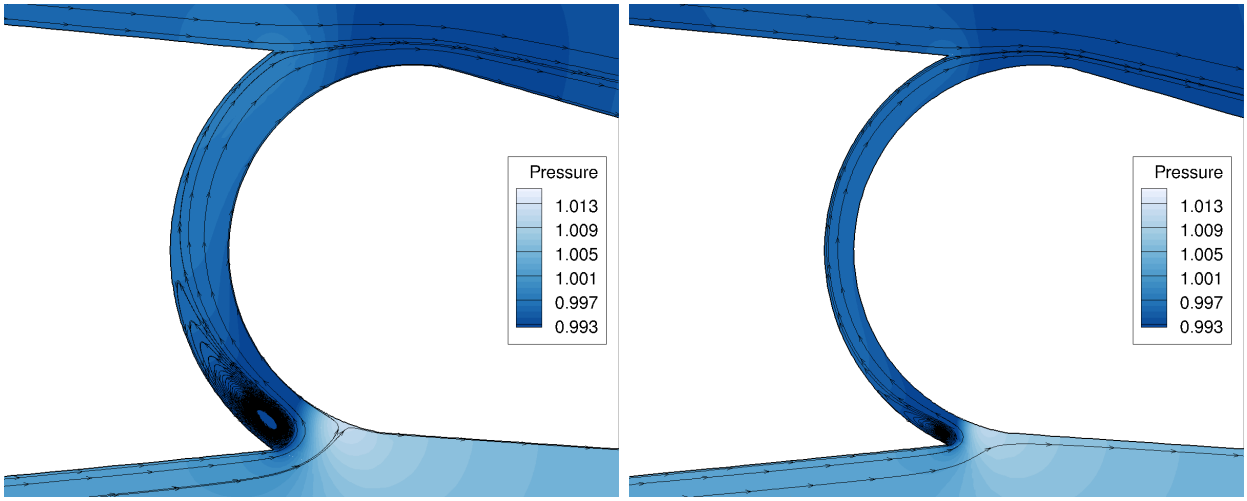


(b) CD versus δ_A



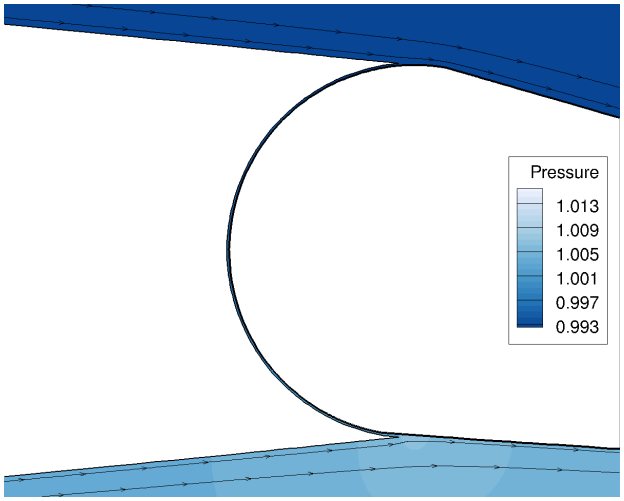
(c) CM versus δ_A

Figure 4.13 Aerodynamics coefficients against the aileron deflection (δ_A) for the five mesh configurations



(a) Large gap

(b) Medium gap



(c) Small gap

Figure 4.14 Pressure contours and streamlines in the gap region for the three gapped configurations

Table 4.6 Aerodynamic coefficients for the three aileron configurations and five mesh techniques

	$\delta_A = 0.0^\circ$			$\delta_A = 5.0^\circ$			$\delta_A = 10.0^\circ$		
	CL	CD [$\times 10^{-4}$]	CM	CL	CD [$\times 10^{-4}$]	CM	CL	CD [$\times 10^{-4}$]	CM
CFL3D	-0.000007	81.9							
Single Grid	-0.000001	83.6	1.37E-07	0.328162	91.7	-5.41E-02	0.635907	104.6	-1.04E-01
large Gap	-0.000244	91.6	2.15E-05	0.242001	114.1	-4.31E-02	0.438474	173.4	-7.82E-02
Medium Gap	0.000069	89.7	-1.39E-05	0.275624	103.0	-4.71E-02	0.523141	139.8	-8.86E-02
small Gap	0.000010	84.2	-2.31E-06	0.323930	88.9	-5.33E-02	0.634720	103.1	-1.04E-01
Collar Grid	-0.000027	83.4	6.43E-06	0.333116	86.5	-5.48E-02	0.641788	112.7	-1.05E-01

CHAPTER 5 CONCLUSION

5.1 Synthesis of work

This thesis presented the extension of the overset method developed inside an in-house CFD solver for the Navier-Stokes equations. It aimed to examine the capacity to handle overlapping meshes that have geometries intersecting with one another. Difficulties inherent to some of the specificities of the flow solver, e.g. a cell-centre formulation and a 9 points stencil for the artificial dissipation, were analyzed and a solution proposed to handle them within the overset method. Through the literature research, the main developments in the world leading groups – such as NASA, Boeing, the French Aerospace Lab (ONERA) and the Deutsches Zentrum fur Luft- und Raumfahrt (DLR) – introduced the concept of collar meshes. These meshes aim at bridging two intersecting geometries with a mesh that partially lies on both geometry. This meshing technique was selected as the method of choice for the current developments.

The first objective of this thesis was to develop a method of simulating complex geometries. To achieve it, three development axes were identified:

1. The need for a better hole cutting algorithm to address concave regions;
2. The need to investigate interpolation in the viscous region of the mesh;
3. The need to properly compute the aerodynamic forces and moments.

These axes have all been studied and treated in chapter 3. A series of verification was performed for each of the methods developed. It has been shown that more work remains, particularly concerning viscous interpolation.

The second objective was to assess the robustness of the method with respect to industrial type applications. Through a demonstration of an transonic case as well as a wing-aileron configuration, the advantages of using the overset method for these types of application was highlighted. Overall, the new implementation of the overset method was able to successfully simulate fluid dynamics problems over complex geometries.

5.1.1 Development

The hole cutting was modified to include a direct cutting approach, rather than the donor-search based hole cutting previously implemented. A Constrained Delaunay Triangulation,

with an advancing front method was chosen to generate an exact mapping of the internal geometry. Coupled with the overlap search already in place in the donor search process, it was shown to be an efficient method to fully discriminate the cells located inside a body. The ONERA has highlighted their tetrahedral hole cutting method – the three-dimensional equivalent of the presented method – as their most accurate and efficient method (Péron, 2016). A consideration must be done however that the generation of a 3D representation using tetrahedral meshes can be tedious, which mitigates the usage of this method. In the current implementation, the information required to generate the triangulation was already available, thus the generation proves relatively simple.

Then, a limitation to the mesh generation process for collar meshes was proposed, to ensure adequate interpolation in the boundary layer region. This forces the collar mesh to have a given discretization near its boundary with the other overlapped meshes. It showed accurate results on Euler and laminar cases, but some issues remain when using these meshes in high Reynolds flow, which is discussed in section 5.2.

Finally, a fully automatic weighted panel method was developed to provide an accurate computation of aerodynamic loads. This implementation followed the methodologies developed by Boger and Dreyer (2006), but was redesigned for the current framework. It uses the information already available from the previous processes to minimize the amount of work done. Overall, this fast classification method ensures that few cells are tested against an overlapped mesh, leaving little place for computation errors.

5.1.2 Applications

The algorithm was tested in a wide range of flow regimes, ranging from Euler and laminar on to fully turbulent high Reynolds steady flows. However, the framework NSCODE offers a wider range of possible applications. It features an ice accretion module, which has already been used with the overset preprocessor (Bourgault-Côté and Laurendeau, 2016). The solver also offers the possibility to solve unsteady, with either a dual time-stepping or Non-Linear Frequency Domain (NLFD) method (Lévesque, 2015). The overset method is particularly suited for unsteady simulations featuring relative motion between different geometries. For example, the case presented in section 4.4 could be used to simulate the unsteady behavior of the aileron and compute the coefficients' derivatives (e.g. $C_{L\dot{\alpha}}$). Some additional work could prove required in order to accurately compute the time derivative terms in the Navier-Stokes equations when using the NLFD method. Literature has shown that when blank cells become computed, the time derivative is improperly computed, thus leading to an incorrect solution (Soucy and Nadarajah, 2009). Finally, the framework also includes a mesh

deformation module using an ALE formulation (Sansas, 2016), which could greatly benefit form the new developments presented in this thesis. Some cases present highly concave regions that become difficult to successfully mesh. Thus, a near-body mesh less dependent on the geometry's curvature and easier to generate would provide more flexibility.

5.2 Limitations of the proposed solution

Although the developments presented in this thesis successfully addressed the objectives defined, some limitations remain when using surface conforming meshes. First, a limitation is imposed on the collar mesh created, to insure a proper interpolation in the viscous layer. Also, the skin friction coefficient remains difficult to properly compute, even when respecting the previous limitation.

5.2.1 Mesh generation

The proposed limitation for the mesh generation of collar mesh, which is to use the background mesh's discretization at the collar boundary, puts a large stress on the mesh generation part of the simulation. Usage of this method in the cases presented showed to the author how much complexity this limitation imposed. Overset meshes were introduced to simplify the mesh generation process, and this limitation brings back a portion of this complexity. Also, this limitation, while doable on two-dimensional cases, is much more difficult to apply three-dimensional configurations. The surface discretization on one geometry might be very different (in terms of number of vertices and spacing) than the intersecting geometry. This would lead to an inconsistency and it would not be possible to generate a structured surface conforming mesh with these constraints. Nonetheless, this has shown the capabilities of the overset method to handle such geometries. As discussed in section 2.4.2, a grid projection algorithm would greatly help to reduce the interpolation error and would alleviate this limitation. Such an algorithm allows for the surface coordinates to be non-confirming between the different meshes. In the current development stage of the overset grid preprocessor, this algorithm is one of most important developments to be included.

5.2.2 Skin friction coefficient

The other limitation encountered is when simulating high Reynolds flows. In these situations, the skin friction coefficient is not properly computed near the collar mesh boundary, leading to a discontinuity in the solution. As the flow properties appear to be continuous on the surface, more analysis is required on the computation of the derivatives of the primitive

variables. This would allow for a better understanding of the challenges brought by the usage of a cell-centre scheme and a dual layer of ghost cells. These specifications of the solver are not much discussed in the overset literature. Ghost cells treatment is also seldom discussed in articles discussing overset aspects. Therefore, the understanding of the current issues are difficult to navigate through, as very few information is available to identify a probable cause.

5.3 Future work

5.3.1 Acceleration

While the new developments included in the overset capacities maintain the robustness of the flow solver, the main acceleration technique to obtain an higher convergence rate remains unavailable with overset meshes. The multigrid technique cannot be used in conjuncture with the overset method in the current framework. This technique accelerates the convergence by computing a solution on a coarser mesh, typically called a level. The coarser levels are usually built by selecting one out of every two vertices. While the overset assembly can be performed on coarser levels without difficulty, it is the solution's update between two levels that is more difficult to properly achieve. Effectively, if the solution on the coarse level is incorrect – such as cells being blanked – it propagates to the finer level and prevents the solver from converging.

5.3.2 Grid analysis

The overset method is primarily a grid treatment. With that in mind, a more detailed analysis of the grids assembled would make an interesting addition to the current preprocessor. The current way of validating the grid assembly almost solely relies on visual inspection. This can be challenging for end-users, particularly in the beginning stage of the learning process. Having a detailed analysis done in an automatic manner would help to ease this step and increase the confidence in the grid assembly. This analysis could potentially include the improvement of mesh metrics of the computed cells, to make sure that poor quality cells are blanked. Furthermore, orphan cells should be identified, because they can prevent convergence of the flow solver. For example, Pointwise offers automatic remediation tools to eliminate these orphan cells (Wyman, 2016). These developments, while not increasing the capacity of the overset preprocessor, aim at providing better information for the user.

5.3.3 Automation of mesh generation

Finally, the automation of mesh generation makes for an interesting research axis. The overset method has allowed engineers to add more and more components to simulations, but also to split them. By doing so, each component presents much less complexity than the global geometry. Efforts in the overset community are focussed on decreasing the time costs of mesh generation, by automating these processes (Chan, 2016). Collar meshes are generally simple surfaces (small curvature, small surface, limited expansion in the volume), which makes them ideal candidates for automation. Moreover, the automation of mesh generation helps dynamic simulations, because the the different time instances are treated without requiring user input.

REFERENCES

BENEK, J. A., STEGER, J. L., DOUGHERTY, F. C., and BUNING, P. G., "chimera: a grid-embedding technique," NASA Ames Research Center, Tech. Rep., Apr. 1986.

BENOIT, C., JEANFAIVRE, G., and CANONNE, E., "Synthesis of onera chimera method developed in the frame of chance program," *ONERA: Tire a Part*, no. 205, p. 1, 2005.

BENOIT, C. and PÉRON, S., "Automatic structured mesh generation around two-dimensional bodies defined by polylines or polyc1 curves," *Computers & Fluids*, vol. 61, pp. 64–76, 2012, doi:10.1016/j.compfluid.2011.09.009.

BLANC, F., "Patch assembly: An automated overlapping grid assembly strategy," *Journal of Aircraft*, vol. 47, no. 1, pp. 110–119, 2010, doi:10.2514/1.44116.

BOGER, D. and DREYER, J., "Prediction of hydrodynamic forces and moments for underwater vehicles using overset grids," in *44th AIAA Aerospace Sciences Meetings*. AIAA Paper 2006-1148, Jan. 2006, doi:10.2514/6.2006-1148.

BOGER, D. A., "User's manual for USURP unique surfaces using ranked polygons," NASA Johnson Space Center, Tech. Rep., Jun. 2006.

BOURGAULT-CÔTÉ, S., "Simulation du givrage sur ailes en flèche par méthodes RANS/Eulérienne quasi stationnaires," Master's thesis, Polytechnique Montréal, Montréal, Apr. 2015.

BOURGAULT-CÔTÉ, S. and LAURENDEAU, E., "Two-dimensional/infinite swept wing ice accretion model," in *53rd AIAA Aerospace Sciences Meeting*. AIAA Paper 2015-0535, 2015, doi:10.2514/6.2015-0535.

BOURGAULT-CÔTÉ, S. and LAURENDEAU, E., "Development of a morphogenetic ice accretion solver using overset grids," in *CFD Society of Canada, 24th annual conference, CFDSC*, 2016.

BOURGAULT-CÔTÉ, S., GHASEMI, S., MOSAHEBI, A., and LAURENDEAU, E., "Extension of a two-dimensional navier–stokes solver for infinite swept flow," *AIAA Journal*, vol. 55, no. 2, pp. 662–667, 2017, doi:10.2514/1.J055139.

BUNING, P. G. and GOMEZ, R. J., "20+ years of chimera grid development for the space shuttle," in *10th Symposium on Overset Composite Grid and Solution Technology*. NASA Langley Research Center, Sep. 2010.

CHAN, W. M., "Enhancements to the hybrid mesh approach to surface loads integration on overset structured grids," in *19th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2009-3990, 2009, doi:10.2514/6.2009-3990.

CHAN, W. M., "Overset grid technology development at nasa ames research center," *Computers & Fluids*, vol. 38, no. 3, pp. 496–503, 2009, doi:10.1016/j.compfluid.2008.06.009.

CHAN, W. M., "Progress in automation of overset structured surface grid generation," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

CHAN, W. M. and BUNING, P., "Zipper grids for force and moment computation on overset grids," in *12th Computational Fluid Dynamics Conference*. AIAA Paper 95-1685-CP, Jun. 1995, doi:10.2514/6.1995-1681.

CHAN, W. M. and PANDYA, S. A., "Chimera grid tools tutorial," in *12th Symposium on Overset Composite Grids and Solution Technology*. NASA Ames Research Center, Oct. 2014.

CHAN, W. M., GOMEZ, R. J., ROGERS, S. E., and BUNING, P. G., "Best practices in overset grid generation," in *32nd AIAA Fluid Dynamics Conference*. AIAA Paper 2002-3191, 2002, doi:10.2514/6.2002-3191.

CHAN, W. M., PANDYA, S., and ROGERS, S. E., "Efficient creation of overset grid hole boundaries and effects of their locations on aerodynamic loads," in *21st AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2013-3074, Jun. 2013, doi:10.2514/6.2013-3074.

CHANDAR, D. D., SITARAMAN, J., and MAVRIPLIS, D. J., "A gpu-based incompressible navier–stokes solver on moving overset grids," *International Journal of Computational Fluid Dynamics*, vol. 27, no. 6-7, pp. 268–282, 2013, doi:10.1080/10618562.2013.829915.

CHENG, S.-W., DEY, T. K., and SHEWCHUK, J., *Delaunay mesh generation*. CRC Press, Dec. 2012.

CHEW, L. P., "Constrained Delaunay triangulations," *Algorithmica*, vol. 4, no. 1-4, pp. 97–108, 1989, doi:10.1007/BF01553881.

CHIU, I. T., "On computations of the integrated space shuttle flowfield using overset grids," Retrospective Thesis and Dissertations, Iowa State University, 1990, paper 9429.

CODER, J., FOSTER, N. F., and BAKER, W. J., "Rotor hub flow simulations using the OVERFLOW solver," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

COOK, P., MCDONALD, M., and FIRMIN, M. C. P., "Airfoil rae2822 pressure distrinution and boundary layer and wake measurements," AGARD AR 138, Tech. Rep., 1979.

DANNENHOFFER, J. and HAIMES, R., "Automated creation of 3-d overset grids directly from solid models," in *20th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2011-3540, Jun. 2011, doi:10.2514/6.2011-3540.

DELOZE, T., "Couplage fluide-solide appliqué à l'étude de mouvement d'une sphère libre dans un tube vertical," Ph.D. dissertation, Université de Strasbourg, 2011.

DELOZE, T., HOARAU, Y., and JAN, D., "Chimera method applied to the simulation of a freely falling cylinder in a channel," *European Journal of Computational Mechanics*, vol. 19, no. 5-7, pp. 575–590, May 2010, doi:10.3166/ejcm.19.575-590.

DEY, T. K., "Delaunay mesh generation of three dimensional domains," *Tessellations in the sciences: Virtues, Techniques and Applications of Geometric Tilings*, 2007.

DRUYOR, C. T., KARMAN, S. L., and JONES, W., "A survey of overset domain assembly methods," in *53rd AIAA Aerospace Sciences Meeting*. AIAA Paper 2015-0910, 2015, p. 0910, doi:10.2514/6.2015-0910.

ENGLISH, R. E., QIU, L., YU, Y., and FEDKIW, R., "Chimera grids for water simulation," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2013, pp. 85–94.

GEA, L. M. and VICKER, D., *CFD Simulation of the Space Shuttle Launch Vehicle with Booster Separation Motor and Reaction Control System Plumes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 233–238, doi:10.1007/3-540-31801-1_30.

GOMEZ, R., "Flowfield of the space shuttle launch vehicule," 2008, [figure]. Consulted 15th of June 2016, from https://commons.wikimedia.org/wiki/File:SSLV_ascent.jpg.

GREGG, R., "The challenges of certification by analysis," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

GREGORY, N. and O'REILLY, C. L., "Low-speed aerodynamic characteristics of naca 0012 aerofoil sections, including the effects of upper-surface roughness simulation hoar frost," NASA, Tech. Rep., Jan. 1970.

HAIMES, R. and DANNENHOFFER, J., "The engineering sketch pad: A solid-modeling, feature-based, web-enabled system for building parametric geometry," in *Fluid Dynamics and Co-located Conferences*. AIAA Paper 2013-3073, Jun. 2013, doi:10.2514/6.2013-3073.

HASANZADEH, K., LAURENDEAU, E., and PARASCHIVOIU, I., "Grid-generation algorithms for complex glaze-ice shapes reynolds-averaged navier–stokes simulations," *AIAA Journal*, vol. 54, no. 3, pp. 847–860, Mar. 2016, doi:10.2514/1.J054076.

HIGGINS, G. J., "An airfoil fitted with a slotted flap," *Journal of the Aeronautical Sciences*, vol. 3, no. 12, pp. 431–433, 1936, doi:10.2514/8.298.

HUE, D., PÉRON, S., WIART, L., ATINAULT, O., GOURNAY, E., RAUD, P., BENOIT, C., and MAYEUR, J., "Validation of a near-body and off-body grid partitioning methodology for aircraft aerodynamic performance prediction," *Computers & Fluids*, vol. 117, pp. 196 – 211, 2015, doi:10.1016/j.compfluid.2015.05.021.

ISHIKAWA, H., TANAKA, K., MAKINO, Y., and YAMAMOTO, K., "Sonic-boom prediction using euler cfd codes with structured/unstructured overset method," in *27th Congress of International Council of the Aeronautical Sciences*, France, Sep. 2010.

JAMESON, A., SCHMIDT, W., and TURKEL, E., "Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes," in *14th Fluids and Plasma Dynamics Conference*. AIAA Paper 1981-1259, Jun. 1981, doi:10.2514/6.1981-1259.

JINGJING, F. and CHAO, Y., "Enhancement and application of overset grid assembly," *Chinese Journal of Aeronautics*, vol. 23, no. 6, pp. 631 – 638, 2010, doi:10.1016/S1000-9361(09)60264-8.

KATZ, A., JAMESON, A., and WISSINK, A., "A multi-solver scheme for viscous flows using adaptive cartesian grids and meshless grid communication," in *47th AIAA Aerospace Sciences Meetings*. AIAA Paper 2009-768, Jan. 2009, doi:10.2514/6.2009-768.

KIM, N. and CHAN, W. M., "Automation of hole-cutting for overset grids using the x-rays approach," in *20th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2011-3052, Jun. 2011, doi:10.2514/6.2011-3052.

KULTAJEV, E., BENOIT, C., PERON, S., and LERAT, A., "Improvement of a bidimensional overset structured mesh generation and adaptation method, based on a near-body/off-body partitioning," in *20th International Meshing Roundtable*, Oct. 2011.

LÉVESQUE, A. T., "Development of an overset structured 2d rans/urans navier-stokes solver using an implicit space and non-linear frequency domain time operators," Master's thesis, Polytechnique Montréal, Montréal, May 2015.

LÉVESQUE, A. T., PIGEON, P., DELOZE, T., and LAURENDEAU, E., "An overset grid 2d/infinite swept wing urans solver using recursive cartesian virtual grid method," *53rd AIAA Aerospace Sciences Meeting*, pp. 5–9, 2015, doi:10.2514/6.2015-0912.

LIAO, W., CAI, J., and TSAI, H. M., "A multigrid overset grid flow solver with implicit hole cutting method," *Computational Methods in Applied Mechanics and Engineering*, vol. 196, pp. 1701–1715, 2007, doi:10.1016/j.cma.2006.09.012.

LIGGETT, N. and SMITH, M. J., "The physics of modeling unsteady flaps with gaps," *Journal of Fluids and Structures*, vol. 38, pp. 255 – 272, 2013, doi:10.1016/j.jfluidstructs.2012.12.010.

MATSUNO, K., YAMAKAWA, M., and SATOFUKA, N., "Overset adaptive-grid method with applications to compressible flows," *Computers & Fluids*, vol. 27, no. 5–6, pp. 599 – 610, 1998, doi:10.1016/S0045-7930(97)00059-5.

MAVRIPLIS, D., "Advancing complex multidisciplinary simulations in the exascale era," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

MEAKIN, R., "Object x-rays for cutting holes in composite overset structured grids," in *Fluid Dynamics and Co-located Conferences*. AIAA Paper 2001-2537, Jun. 2001, doi:10.2514/6.2001-2537.

MILLER, S., CAMPBELL, R., ELSWORTH, C., PITT, J., and BOGER, D., "An overset grid method for fluid-structure interaction," *World Journal of Mechanics*, vol. 4, pp. 217–237, 2014, doi:10.4236/wjm.2014.47023.

MOSAHEBI, A. and LAURENDEAU, E., "Convergence characteristics of fully and loosely coupled numerical approaches for transition models," *AIAA Journal*, vol. 53, no. 5, pp. 1399–1404, 2015, doi:10.2514/1.J053722.

NOACK, R. W., "A direct cut approach for overset hole cutting," in *18th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2007-3835, Jun. 2007, doi:10.2514/6.2007-3835.

NOACK, R. W., "On overset hole cutting and the problems encountered," in *13th Symposium on Overset Composite Grids and Solution Technology*, Oct. 2016.

NOACK, R. W., BOGER, D. A., KUNZ, R. F., and CARRICA, P. M., "SUGGAR++: An improved general overset grid assembly capability," in *19th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2009-3992, Jun. 2009, doi:10.2514/6.2009-3992.

PANDYA, M., FRINK, N., and NOACK, R., "Progress toward overset-grid moving body capability for usm3d unstructured flow solver," in *17th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2005-5118, Jun. 2005, doi:10.2514/6.2005-5118.

PANDYA, S., CHAN, W. M., and KLESS, J., "Automation of structured overset mesh generation for rocket geometries," in *19th AIAA Computational Fluid Dynamics*. AIAA Paper 2009-3993, Jun. 2009, doi:10.2514/6.2009-3993.

PARENTEAU, M., DEMERS, F., and LAURENDEAU, E., "Numerical implementation of a plasma actuator in a RANS flow solver coupled with the $\gamma - Re_\theta$ transition model," in *CFD Society of Canada, 23rd annual conference, CFDSC*, 2015.

PÉRON, S., BENOIT, C., LANDIER, S., and RAUD, P., "Cassiopée - CFD Advanced Set of services in an Open Python EnvironnEment," in *12th Symposium on Overset Composite Grids and Solution Technology*. Onera, Oct. 2014.

PÉRON, S., "A review of overset grid technology at ONERA," in *13th Symposium on Overset Composite Grids and Solution Technology*. Onera, Oct. 2016.

PIGEON, A., LEVESQUE, A. T., and LAURENDEAU, E., "Two-dimensional navier–stokes flow solver developed at ecole polytechnique de montreal," in *CFD Society of Canada, 22nd annual conference, CFDSC*, 2014.

PIGEON, A., "Développement d'une méthode d'accélération par grilles virtuelles récursives pour l'assemblage de maillages chimères," Master's thesis, Polytechnique Montréal, Montréal, 2015.

PINAEV, D., SCHAFER, F., and SCHRECK, E., "Multi-disciplinary applications using overset grid technology in STAR-CCM+," in *12th Symposium on Overset Composite Grids and Solution Technology*. CD-adapco, Oct. 2014.

ROBITAILLE, M., MOSAHEBI, A., and ÉRIC LAURENDEAU, "Design of adaptive transonic laminar airfoils using the transition model," *Aerospace Science and Technology*, vol. 46, pp. 60 – 71, 2015, doi:http://dx.doi.org/10.1016/j.ast.2015.06.027.

ROGERS, S., ROTH, K., NASH, S., BAKER, M., SLOTNICK, J., WHITLOCK, M., and CAO, H., "Advances in overset cfd processes applied to subsonic high-lift aircraft," in *18th AIAA Applied Aerodynamics conference*. American Institute of Aeronautics and Astronautics, Aug. 2000, doi:10.2514/6.2000-4216.

ROGET, B. and SITARAMAN, J., "Robust and efficient overset grid assembly for partitioned unstructured meshes," *Journal of Computational Physics*, vol. 260, pp. 1 – 24, 2014, doi:10.1016/j.jcp.2013.12.021.

RUMSEY, C., "Turbulence modeling resource," NASA Langley Research Center, 2016, page consulted on 10 February 2017. [Online]. Available: https://turbmodels.larc.nasa.gov/naca0012_val.html

SANSAS, F., "Simulation numérique de la dynamique d'une structure élancée flexible dans un écoulement de fluide compressible," Master's thesis, Polytechnique Montréal, Montréal, Feb. 2016.

SCHWARZ, T., SPIERING, F., and KROLL, N., "Grid coupling by means of Chimera interpolation techniques," in *2nd Symposium on "Simulation of Wing and Nacelle Stall"*, Jun. 2010.

SCLAFANI, A. J., VASSBERG, J. C., HARRISON, N. A., RUMSEY, C. L., RIVERS, S. M., and MORRISON, J. H., "Cfl3d/overflow results for dlr-f6 wing/body and drag prediction workshop wing," *Journal of Aircraft*, vol. 45, no. 3, pp. 762–780, May 2008, doi:10.2514/1.30571.

SOUCY, O. and NADARAJAH, S., "A nlfd method for the simulation of periodic unsteady flows for overset meshes," in *19th AIAA Computational Fluid Dynamics*. AIAA Paper 2009-4271, Jun. 2009, doi:10.2514/6.2009-4271.

STEGER, J. L., DOUGHERTY, F. C., and BENEK, J. A., "A chimera grid scheme.[multiple overset body-conforming mesh system for finite difference adaptation to complex aircraft configurations]," in *Advances in Grid Generation*. American Society of Mechanical Engineering, 1983, pp. 59–69.

SUHS, N. E., ROGERS, S. E., DIETZ, W. E., and KWAK, D., "Pegasus 5: An automated pre-processor for overset-grid cfd," NASA, Tech. Rep., 2002.

SWANSON, R. C. and LANGER, S., "Comparison of naca 0012 laminar flow solutions: Structured and unstructured grid methods," NASA and the DLR, Tech. Rep., Jan. 2016.

SWANSON, R. and TURKEL, E., "On central-difference and upwind schemes," *Journal of Computational Physics*, vol. 101, no. 2, pp. 292 – 306, 1992, doi:10.1016/0021-9991(92)90007-L.

TOGASHI, F., ITO, Y., NAKAHASHI, K., and OBAYASHI, S., "Extensions of overset unstructured grids to multiple bodies in contact," *Journal of aircraft*, vol. 43, no. 1, pp. 52–57, 2006, doi:10.2514/1.540.

TOGASHI, F., ITO, Y., NAKAHASHI, K., and OBAYASHI, S., "Overset unstructured grids method for viscous flow computations," *AIAA journal*, vol. 44, no. 7, pp. 1617–1623, 2006, doi:10.2514/1.4292.

VASSBERG, J. C. and JAMESON, A., "In pursuit of grid convergence for two-dimensional euler solutions," *Journal of Aircraft*, vol. 47, no. 4, pp. 1152–1166, 2010, doi:10.2514/1.46737.

VERSTEEG, H. K. and MALALASEKERA, W., *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

WANG, Z. J. and PARTHASARATHY, V., "A fully automated chimera methodology for multiple moving body problems," *International Journal for Numerical Methods in Fluids*, vol. 33, no. 7, pp. 919–938, 2000, doi:10.1002/1097-0363(20000815)33:7<919::AID-FLD944>3.0.CO;2-G.

WIGTON, L., "Polymixsur, boeing's replacement for mixsur," in *Proceedings of the 7th Symposium on Overset Composite Grid and Solution Technology, Huntington Beach, California*, 2004.

WISSINK, A., "Recent advances in the CREATE-AV Helios rotorcraft simulation code," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

WYMAN, n., "Integrated overset mesh generation and assembly with pointwise," in *12th Symposium on Overset Composite Grids and Solution Technology*. Pointwise, Oct. 2014.

WYMAN, N., "Pointwise integrated overset meshing and assembly," in *13th Symposium on Overset Composite Grid and Solution Technology*, Oct. 2016.

XUEFEI, L., YUAN, L., and ZHANSEN, Q., "2014 asia-pacific international symposium on aerospace technology, apisat2014 september 24-26, 2014 shanghai, china applications of overset grid technique to cfd simulation of high mach number multi-body

interaction/separation flow," *Procedia Engineering*, vol. 99, pp. 458 – 476, Sep. 2015, doi:10.1016/j.proeng.2014.12.560.

ZHANG, H., FAN, J., YUAN, W., and ZHANG, W., "Structure overset grid method and its applications to simulation of multi-body separation," *Science China Physics, Mechanics & Astronomy*, vol. 58, no. 9, pp. 1–8, 2015, doi:10.1007/s11433-015-5689-0.

APPENDIX A Wing-aileron configuration

Here is presented some additional figures regarding the steady wing-aileron configuration study presented in section 4.4. The figures included in this appendix are for the benefit of the reader, as the most important figures have already been shown.

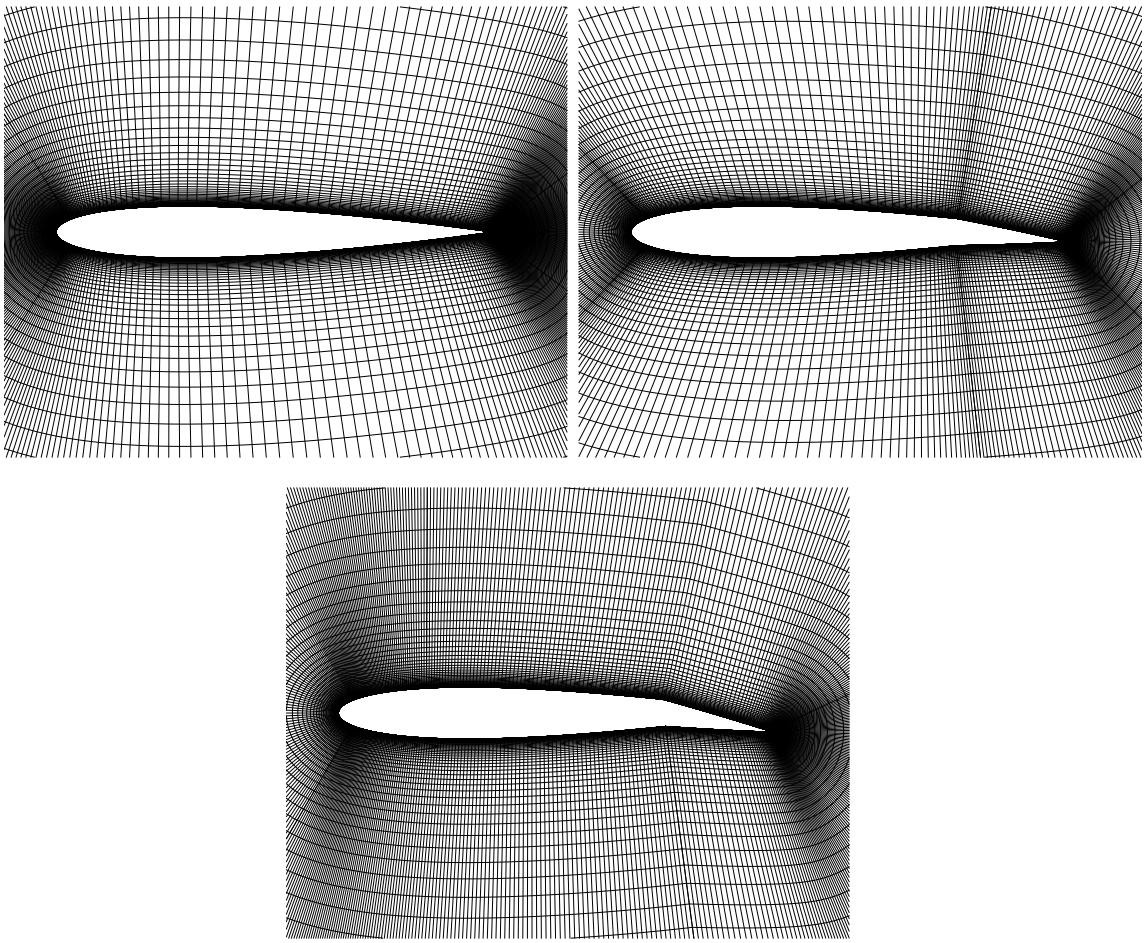
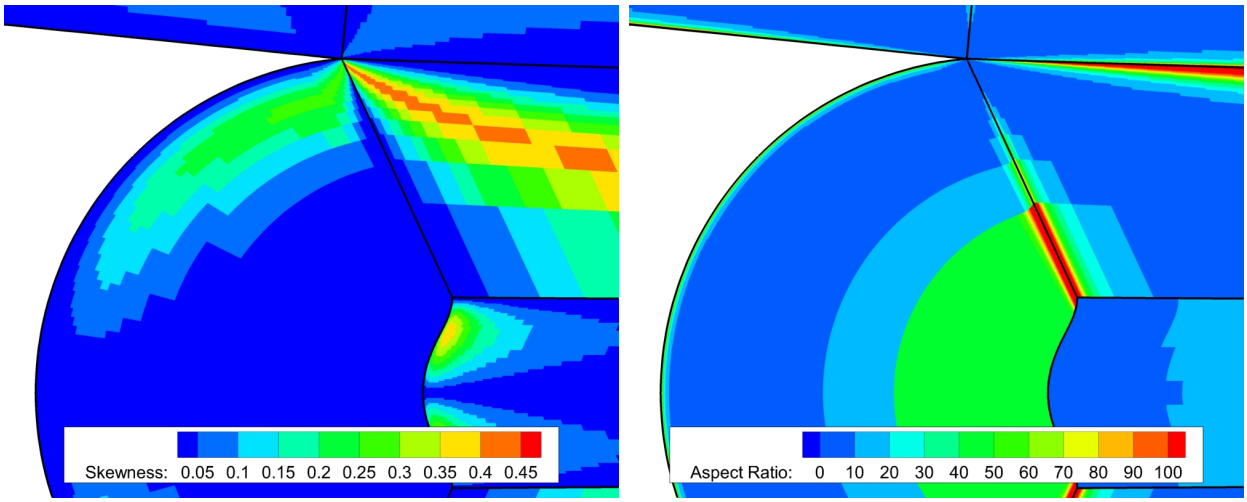
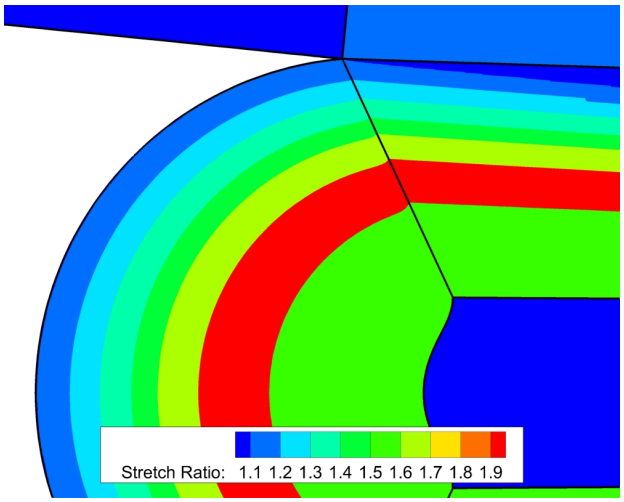


Figure A.1 Single grid used for the three aileron deflection angles



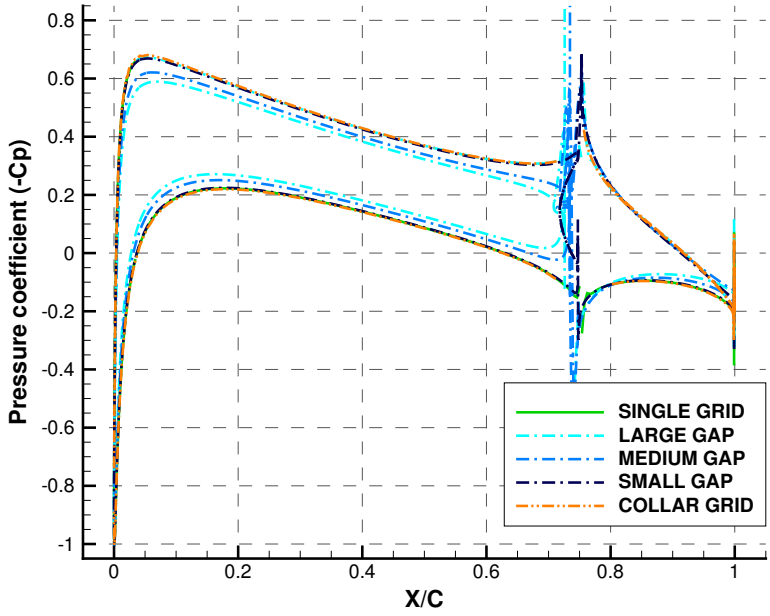
(a) Skewness

(b) Aspect ratio

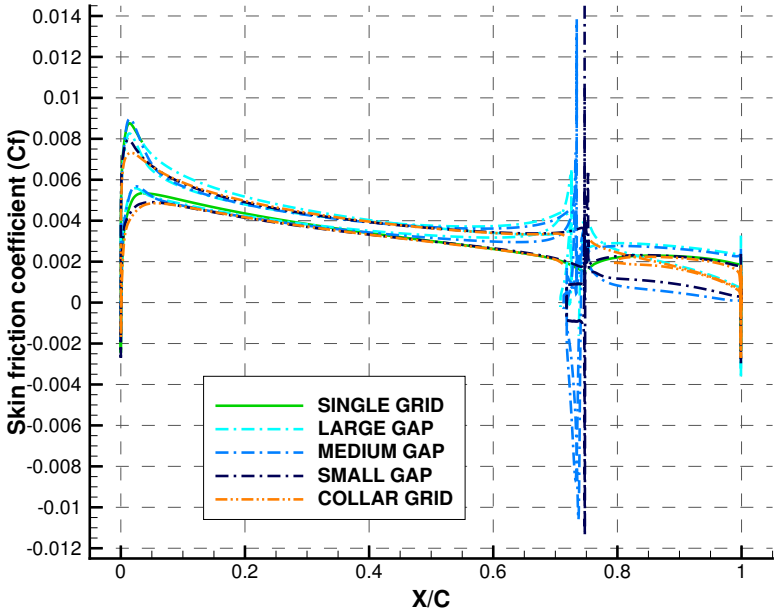


(c) Stretching ratio

Figure A.2 Metrics of the dual C-type mesh generated for the small gap wing mesh in the aileron region

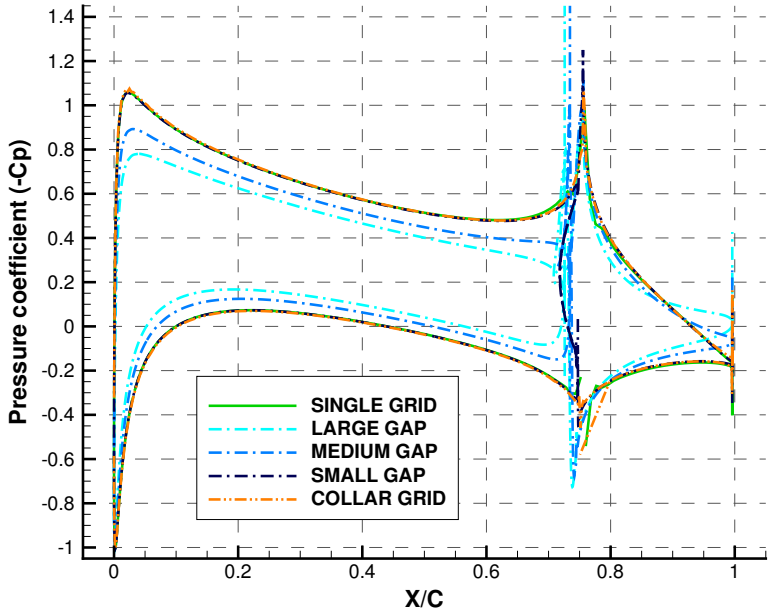


(a) Pressure coefficient

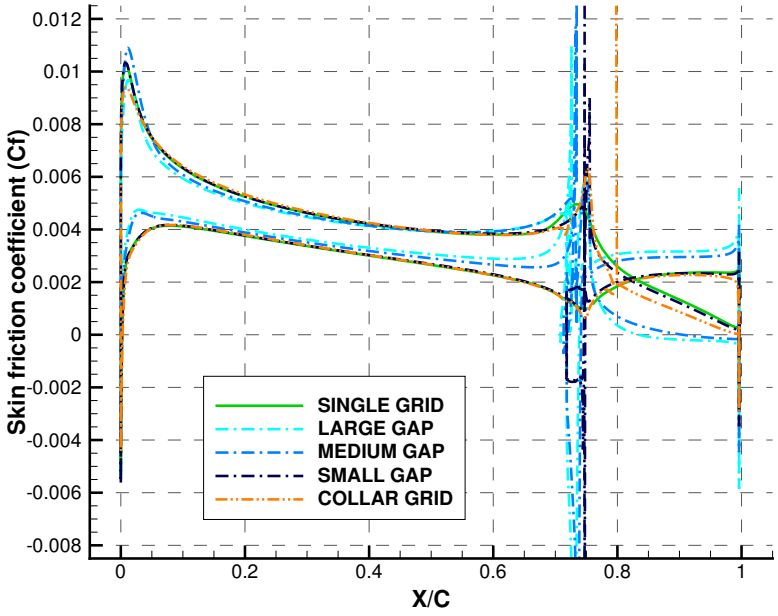


(b) Skin friction coefficient

Figure A.3 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 5.0˚

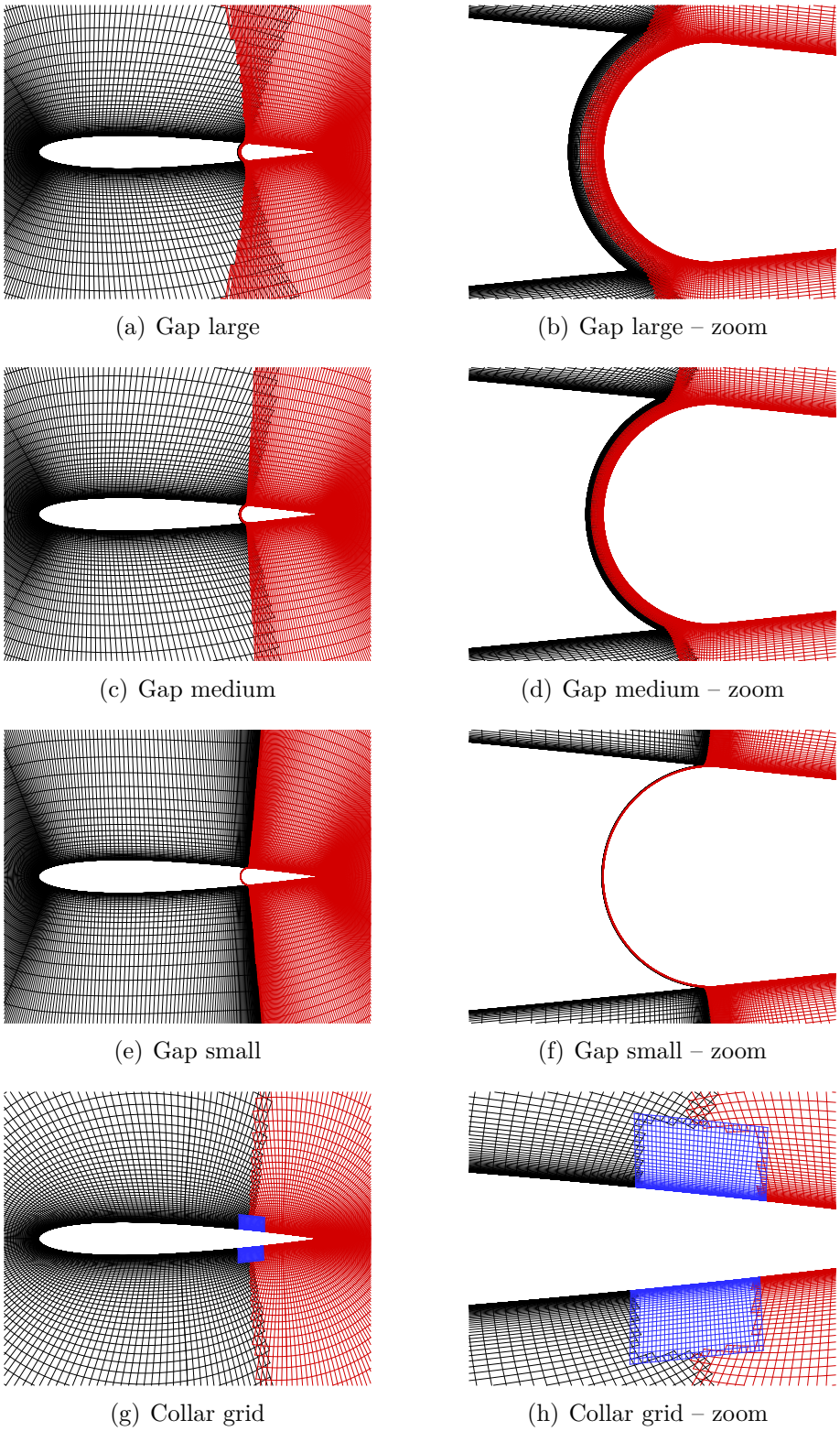


(a) Pressure coefficient



(b) Skin friction coefficient

Figure A.4 Pressure coefficient (a) and skin friction (b) coefficient for the five mesh configurations at the aileron deflection of 10.0˚



(a) Gap large

(b) Gap large – zoom

(c) Gap medium

(d) Gap medium – zoom

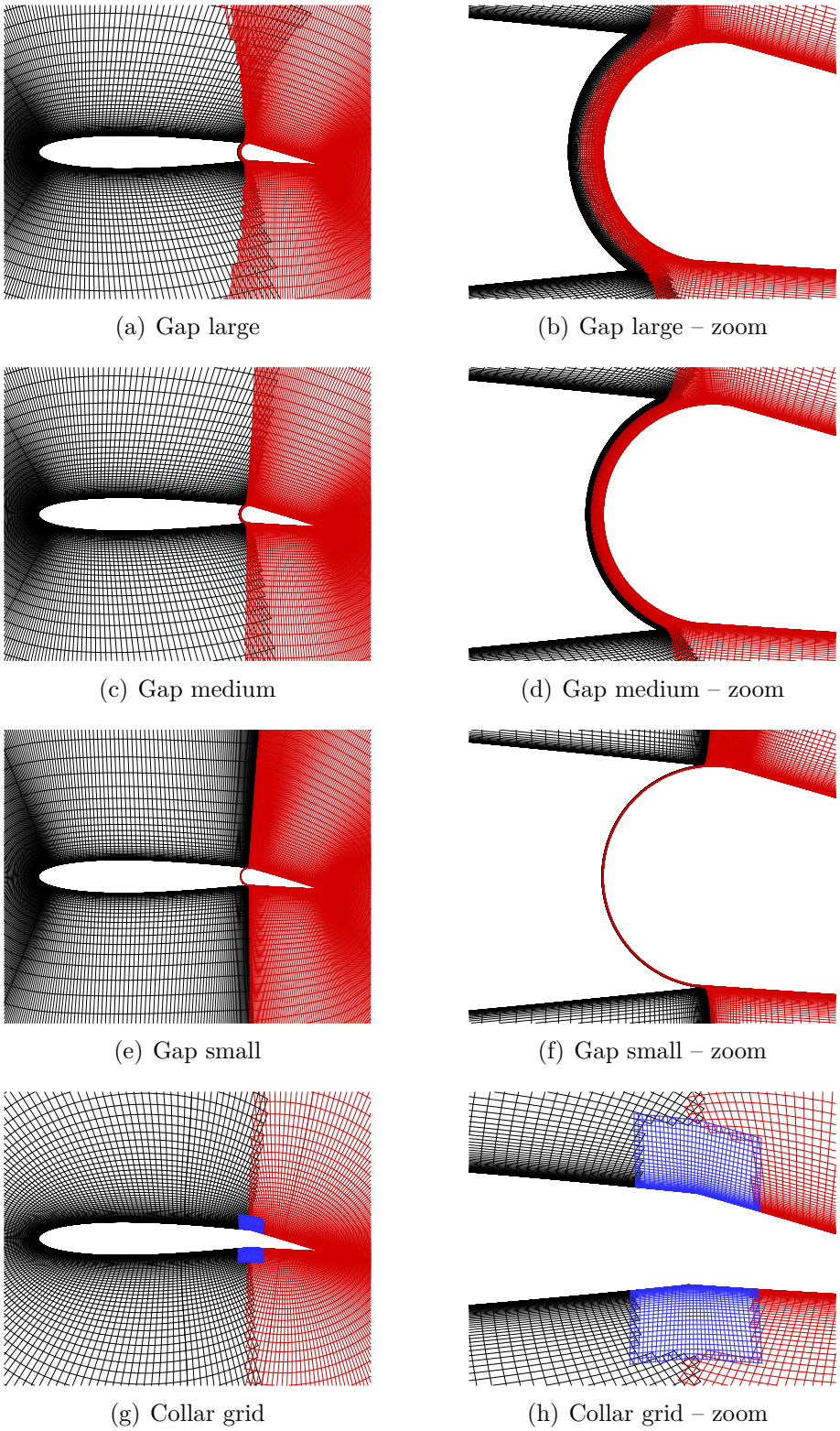
(e) Gap small

(f) Gap small – zoom

(g) Collar grid

(h) Collar grid – zoom

Figure A.5 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 0.0°



(a) Gap large

(b) Gap large – zoom

(c) Gap medium

(d) Gap medium – zoom

(e) Gap small

(f) Gap small – zoom

(g) Collar grid

(h) Collar grid – zoom

Figure A.6 Grid assemblies on the three gap configurations and the collar grid configuration for the aileron deflection of 10.0°